A New Traffic Control Scheme for ATM Networks

by

Sabih Masarrat Alsayed

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER ENGINEERING

July, 1996

A New Traffic Control Scheme for ATM Networks

BY

Sabih Masarrat Alsayed

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

Computer Engineering

July 1996

UMI Number: 1379308

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
DHAHRAN 31261, SAUDI ARABIA

COLLEGE OF GRADUATE STUDIES

This thesis, written by **Sabih Masarrat Alsayed** under the direction of his Thesis Advisor and approved by his Thesis Committee, has been presented to and accepted by the Dean of the College of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER ENGINEERING

Thesis Committee

Dr. Habib Youssef (*Thesis Advisor*)

Dr. Muhammad S. T. Benten (*Member*)

Sadiq Sait . M

Dr. Sadiq M. Sait (*Member*)

Dr. Samir H. Abdul Jauwad (*Member*)

Dr. Muhammad S. T. Benten
(*Department Chairman*)

Dr. Ala H. Al-Rabeh
(*Dean, College of Graduate Studies*)

17.4.96
Date

Dedicated to

my parents

Acknowledgment

First and foremost, all praise to the Almighty Allah Who gave me the courage and patience to carry out this work. I am happy to have had a chance to glorify His name in the sincerest way through this small accomplishment and ask Him to accept my efforts and forgive my shortcomings. May He guide me and the whole humanity to the right path *(Aameen)*.

Acknowledgement is due to King Fahd University of Petroleum and Minerals for providing support to this work.

My deep appreciation goes to my thesis advisor Dr. Habib Youssef for his constant help, guidance and the attention that he devoted throughout the course of this work . His priceless suggestions made this work interesting and learning for me. He was always kind, understanding and sympathetic to me. Working with him was indeed a wonderful and learning experience which I thoroughly enjoyed. May Allah give him excellence and prosperity in his career and grant him a place in Heaven in the hereafter.

Thanks are also due to my thesis committee members Dr. Muhammad S. T. Benten, Dr. Sadiq M. Sait, and Dr. Samir H. Abdul Jauwad for their interest, cooperation, advice and constructive criticism. May Allah give them prosperity in their lives and grant them a place in Heaven in the hereafter.

I would also like to thank my brothers and sister who always gave me a loving, enjoyable and cherished company all my life. May Allah give them joy and satisfaction in their lives and grant them a place in Heaven in the hereafter.

Lastly, but not the least, I would like to thank all my friends at KFUPM for their good company. But special thanks to my friends at 730; Atif, Aamir, Mansoor, Ibrahim, Aashiq, Kamran, Piracha, and Asim, who made my stay at KFUPM joyous and memorable. May Allah fulfil the ambitions of their lives and grant them a place in Heaven in the hereafter.

Contents

List of Figures

List of Tables

Abstract

Name: Sabih Masarrat Alsayed

Title: A New Traffic Control Scheme for ATM Networks

Major Field: Computer Engineering

Date of Degree: July 1996

Traffic control is an important issue in data networks. The problem has been extensively studied for wide-area computer networks. But the solutions appropriate for WANs have not been found efficient for Asynchronous Transfer Mode (ATM) networks, which are based on high bandwidth optical fibers. In ATM networks, the problem is further complicated due to the statistical multiplexing capability of these networks. In this thesis a new traffic control scheme, based on reactive congestion control principle but with a fast response time, has been designed. A simulator is developed to evaluate the performance of the proposed scheme under various load conditions. The scheme is evaluated with respect to two classes of performance metrics: (1) quantitative, namely cell loss and utilization; and (2) qualitative, namely efficiency, fairness, distributedness, and convergence. The proposed scheme is also compared with the Leaky Bucket congestion control scheme. Experimental results indicate that the scheme is very effective in swiftly reacting to congestion. Furthermore, the new scheme showed a marked improvement over the leaky bucket scheme.

Master of Science Degree
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia
July, 1996

خلاصة الرسالة

الاسم : صبيح مسرت السيد
عنوان الرسالة : طريقة جديدة للسيطرة على الحركة المرورية في شبكات التبادل غيرالمتزامن ATM
التخصيص : هندسة الحاسب الآلي
تاريخ الشهادة : يوليو ١٩٩٦م

ان السيطرة على الحركة المرورية لكميات المعلومات وتبادلها عبر الشبكات المعلوماتية من أهم الامور و أصعبها. ولقد درست هذه المشكلة فى شبكات الحاسب الآلى ذات النطاق الواسع. غير ان الحلول في ذلك المجال لم تثبت فاعلية في فجال شبكات ال (ATM) "شبكات التبادل غير المتزامن". وسبب ذلك كون الوسط الفيزياىء الناقل "الالياف الضوئية " في هذه الشبكات ذات قدرة استيعابية عالية لكم أكبر من المعلومات. بالاضافة الى امكانيات التداخل الاحصائى في هذه الشبكات. ولقد قمنا في هذه الاطروحة بتصميم طريقة جديدة للتحكم في الحركة المرورية معتمدين علي مبدأ التحكم التفاعلي بالاختناقات المرورية في الشبكة، مع التركيز على تحقيق استجابة سريعة لمتغيرات الشبكة. كما قمنا ببناء نظام محاكاة لتقييم هذه الطريقة الجديدة من الحيثين الكمي والنوعي. فمن الناحية الكمية قمنا بقياس عدد الخلايا المنقولة عبر الشبكة و عدد الخلايا المستغلة. وأما من الناحية النوعية فقد قمنا بقياس فاعلية الطريقة، وعدالة توزيع الخلايا في الشبكة، و قدرة الشبكة علي الوصول الي حالة متزنة باستعمال هذه الطريقة. كما قمنا بدراسة مقارنة لهذه الطريقة مع طريقة التحكم بالحركة المرورية المعروفة باسم "الدلو المتسربة". ولقد اثبتت النتائج أن هذه الطريقة الجديدة كانت انجع وافضل من طريقة "الدلو المتسربة".

درجة الماجستير فى العلوم
جامعة الملك فهد للبترول و المعادن
الظهران - المملكة العربية السعودية
يوليو ١٩٩٦م

Chapter 1

Introduction

Wide-area computer communication networks usually have a large number of nodes which have different transmission capacities. Moreover, the processing powers of the sources attached to these nodes can come from a wide range of processors. This heterogeneity gives rise to two important problems namely *congestion control* and *flow control.* **Congestion** takes place at a node if it cannot transmit packets at the same rate at which it receives them from the sources and/or from other nodes attached to it. The packets start accumulating at the node and eventually are lost when the buffer space is exhausted. To avoid this situation some mechanism is incorporated in the networking software (or hardware) which is called the congestion control mechanism. **Flow control**, like congestion control, is a mechanism which is used to control a slightly different situation. A destination node with lesser processing power (speed) might be flooded with packets from a relatively higher powered source node. This situation will also result in the loss of packets when the buffer capacity of the destination node is exhausted. Flow control mechanism is used to adjust the flow rate of the source with respect to the buffer size and the processing capacity of the destination node. The two problems are strongly interrelated and are sometimes solved as one problem [1]. Congestion and flow control are also called *Traffic Control.*

Asynchronous Transfer Mode (ATM) is the switching technology used in the Broadband Integrated Services Digital Networks (B-ISDN) to provide all types of communication services to the end users [2]. ITU-T (CCITT) Recommendation I.121 [3] presents an overview of B-ISDN capabilities:

> B-ISDN supports switched, semi-permanent and permanent, point-to-point and point-to-multipoint connections and provides on demand, reserved and permanent services. Connections in B-ISDN support both circuit mode and packet mode services of a mono- and/or multi-media type, of a connectionless or connection-oriented nature, and in a bidirectional or unidirectional configuration.
>
> A B-ISDN will contain intelligent capabilities for the purpose of providing advanced service characteristics, supporting powerful operation and maintenance tools, network control and management.

The basic unit of information in ATM is a fixed size packet of 53 octets (bytes) called a *"cell"*. It employs flexible allocation of bandwidth to allocate the available transmission bandwidth to different sources. The most important feature of ATM is *statistical multiplexing* (discussed in the next section) of the traffic from different sources. An ATM based B-ISDN network consists of many nodes which can have different transmission and processing (switching) capabilities. This makes an ATM network similar to the wide area packet network. Therefore, B-ISDN networks also have the inherent problems of congestion and flow control, or traffic control.

1.1 Statistical Multiplexing

The main objective of statistical multiplexing is to make use of all the available bandwidth by multiplexing the connections to various sources asynchronously. It is based on the fact that many types of individual connections tend to be variable bit rate traffic sources. Instead of allocating specific time slots to the sources, a robust scheduler is used to quickly transport the incoming cells whenever the bandwidth becomes available. This concept allows the maximum bit rate supported by the network at any time to be less than the sum of the peak bit rates of the various traffic sources. Sometimes, the cells on certain connections may be of a higher priority, in which case the scheduler gives those cells priority in allocating the available bandwidth. If the connections require low cell loss probabilities the scheduler will have to allocate a higher bandwidth to those connections.

1.2 Quality of Service (QoS)

A service is essentially a request to transport a type of traffic (data, voice, video). Each type of traffic has its own characteristics specified in the form of lower limits with respect to the traffic parameters. The service which provides the required limits to a traffic type is said to have the required *Quality of Service* (QoS). The QoS provided by the network can be described at the **call level** as well as the **cell level**.

Some parameters that are used in describing QoS at the *call level* are:

- Connection acceptance probability, and
- Connection setup delay.

The following parameters are used to describe QoS at the *cell level*[1]:

- Cell transfer delay,
- Cell delay jitter,
- Cell loss probability, and
- Bit error rate.

[1] The definition of these parameters follows later in the section

The QoS specified at the call level is used to determine whether or not the connection can be established. The QoS at the cell level, however, is mainly used to determine the amount of delay and data loss that the user can tolerate in the delivery of data to the destination after the establishment of the connection. QoS at the call level is handled by the Control Plane (Figure 1.4) using complex algorithms. QoS at the cell level is handled by the ATM layer (Figure 1.4). For the purpose of this thesis, we will focus on the QoS at the *cell level*, since it pertains to controlling cell traffic after the connection has been established.

1.3 ATM Networking Framework

ATM is a connection-oriented cell switching technique. The connections within the ATM layer are called *virtual connections.* These virtual connections have a two-level hierarchy:

- Virtual channel level, and
- Virtual path level.

These are defined in ITU-T (CCITT) Recommendation I.113 [4] as:

Virtual Channel (VC): "A concept used to describe unidirectional transport of ATM cells associated with a common unique identifier value". This identifier is called the virtual channel identifier (VCI) and is part of the cell header. The *Virtual*

Channel Connection (VCC) is defined in ITU-T (CCITT) Recommendation I.113 [4] as follows:

> "A concatenation of virtual channel links that extends between two points where the adaptation layer is accessed."

To establish a VCC at the B-ISDN UNI, one of the following three methods can be used [2]:

1. Semi-permanent or permanent VCCs are established during the subscription time. No signalling procedure is necessary.
2. Establishment of a switched end-to-end VCC can be done by a user-to-network signalling procedure.
3. If a Virtual Path Connection (VPC)(defined below) between two B-ISDN UNIs already exists, a VCC within this VPC can be established by employing a user-to-user signalling procedure.

The cell sequence integrity is preserved within a VCC. Traffic parameters are individually negotiated at VCC establishment between the user and the network. All cells originating from the user are monitored by the network in order to ensure that the agreed upon parameters are not violated. A VCC can be switched at a VC switch as shown in Figure 1.1. This type of switching takes place when a VP

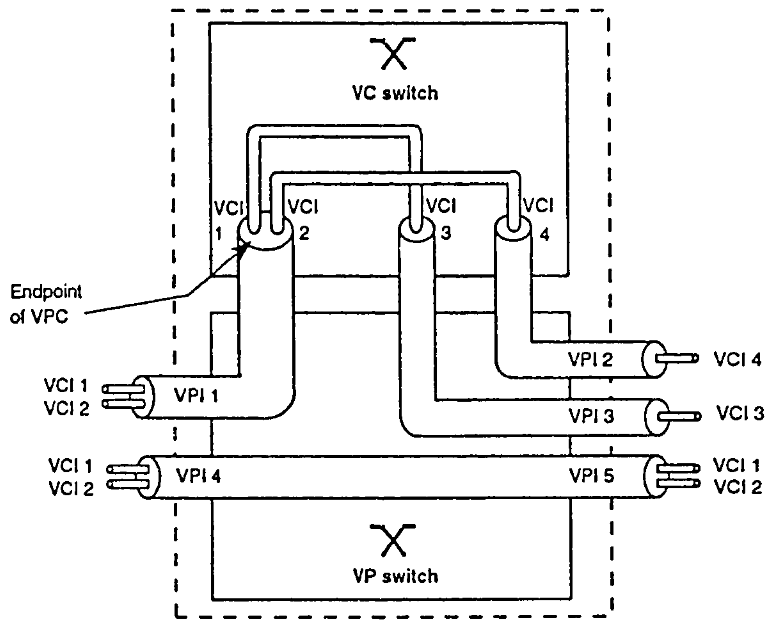

Figure 1.1: Virtual Circuit (VC) switching.

(described below) terminates at a node which is not the destination node for the traffic carried on the VCC.

At VCC establishment the QoS for the connection is negotiated between the user and the network. The agreement reached with respect to QoS defines a traffic contract. A traffic contract cannot be changed for the duration of the connection. In general, the value of the VCI field is independent of the service provided over that VC.

Virtual Path (VP): 'A concept used to describe unidirectional transport of cells belonging to virtual channels that are associated by a common identifier value'. This identifier is called the virtual path identifier (VPI) and is also part of the cell header.

The definition of the *Virtual Path Connection* (VPC) is given in ITU-T (CCITT) Recommendation I.113 [4]:

> "A concatenation of virtual path links that extends between the point where the virtual channel identifier values are assigned and the point where those values are translated or removed."

The methods for establishing a VPC between VPC endpoints are not yet defined but they may obey one of the following principles:

1. A VPC is established on a subscription basis and therefore no signalling procedure is necessary.

2. The VPC establishment may be controlled by the customer. For these purposes, signalling or network management procedures are utilized.

3. A VPC can also be established by the network using network signalling procedures.

Within a VPC the cell sequence integrity is preserved for each VCC carried. During VPC establishment the traffic parameters for the VPC are negotiated between the user and the network. All input cells from the user to the network are monitored to supervise the traffic parameters. Switching of VP is shown in Figure 1.2. Note that a particular node can have several VC's with the same VCI value

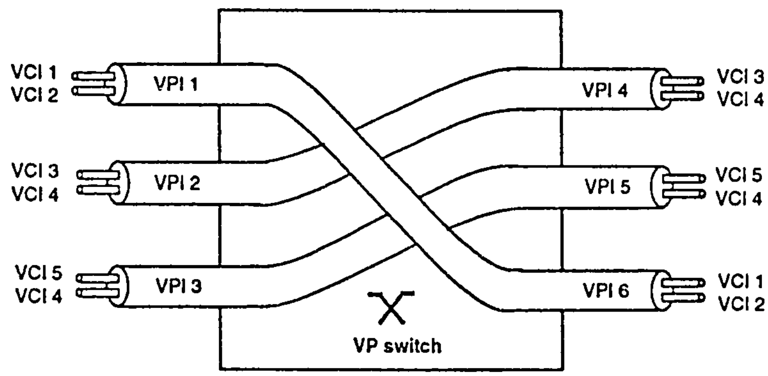

Figure 1.2: Virtual Path (VP) switching.

provided that they are in different VP's. A node cannot have more than one VP with the same VPI.

At VPC establishment the QoS is determined from a number of QoS classes supported by the network. The QoS class associated with a VPC cannot be changed during the duration of the VPC. A VPC can carry VCCs requiring different QoS classes. Therefore, the QoS of the VPC must meet the most demanding QoS of the VCCs carried.

Figure 1.3 demonstrates the relationship between virtual channel, virtual path and transmission path: a transmission path may comprise several virtual paths and each virtual path may carry several virtual channels. The virtual path concept allows grouping of several virtual channels with similar traffic characteristics.

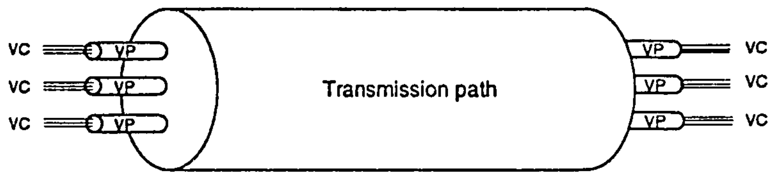

Figure 1.3: Relationship between virtual channel, virtual path and transmission path.

ATM Service Classes

ATM UNI Specification [5] has defined the following four Service Classes which are supported by an ATM network.

- Service Class A: Circuit emulation, Constant bit rate video
- Service Class B: Variable bit rate (VBR) audio and video
- Service Class C: Connection-oriented data transfer
- Service Class D: Connectionless data transfer

As part of the connection setup process, the user chooses one of the specified QoS class that is appropriate for the given application. Once the connection is established, it is the responsibility of the ATM layer to provide the required QoS by ensuring that the parameters for the specified QoS are maintained at or below their objective values.

1.3.1 ATM Layered Model

The ISO OSI model has been used successfully to model all sorts of communication systems. The same logical hierarchical architecture is used for the ATM B-ISDN network in ITU-T (CCITT) Recommendation I.321 [6]. The model also uses the concept of separated planes for the segregation of user, control and management functions. This "plane" approach was already used in (Narrowband) ISDN and is described in ITU-T (CCITT) Recommendation I.320 [7, 8], which contains the ISDN Protocol Reference Model (PRM).

The B-ISDN protocol model for ATM is shown in Figure 1.4. It contains three planes: a *user plane* to transport user information, a *control plane* mainly composed of signalling information, and a *management plane*, used to maintain the network and to perform operational functions. In addition, a third dimension is added to the PRM, called the plane management, which is responsible for the management of the different planes.

For each plane, a layered approach, as in OSI, is used. ITU-T has not defined the relation between the layers of the BISDN ATM protocol model and those of the OSI model. The following relations can, however, be found; the physical layer is more or less equivalent to the Physical layer of the OSI model, and mainly performs functions on the bit level. The ATM layer can be located at the lower edge of the Data Link layer of the OSI model. The adaptation layer performs the adaptation

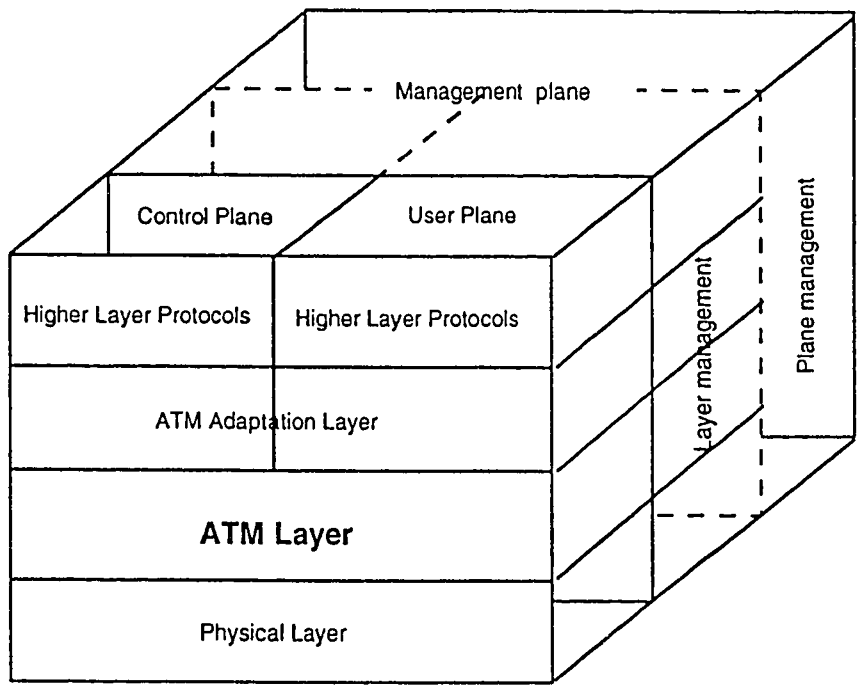

Figure 1.4: B-ISDN ATM protocol reference model.

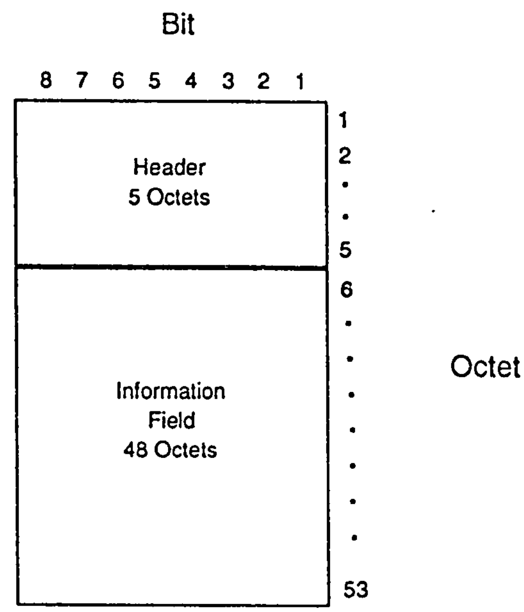

Figure 1.5: ATM cell structure.

of higher layer protocols, be it signalling or user information, to the fixed size ATM cells.

1.3.2 ATM Cell Structure and Header Fields

In ITU-T (CCITT) Recommendation I.361 [9] the format of the ATM cells is described in detail. The cell structure which was finally selected by ITU-T (CCITT) contains a 48 octet (byte) information field and a 5 octet header (Figure 1.5). The octets are sent in an increasing order, starting with octet 1 of the header. Within an octet, the bits are sent in a decreasing order, starting with bit 8. For all fields of an ATM cell, the first bit sent is also the Most Significant Bit (MSB).

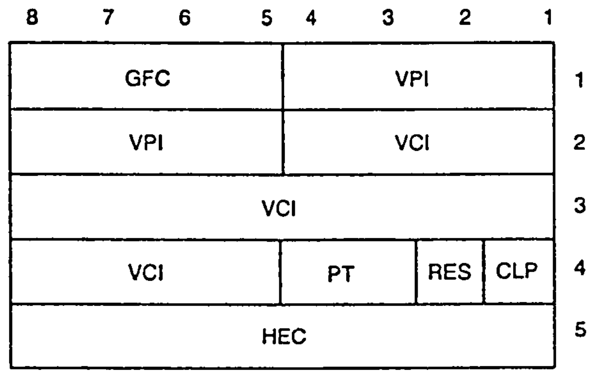

Figure 1.6: ATM header structure at UNI.

At the User Network Interface (UNI), the header structure is shown in Figure 1.6. The first field contains 4 bits for the Generic Flow Control (GFC). However, coding and functionality of the GFC bits is not yet specified by ITU-T. The second field is the routing field, subdivided in a Virtual Channel Identifier (VCI) field of 16 bits and a Virtual Path Identifier (VPI) field of 8 bits. The Payload Type (PT) field is coded in 2 bits with 00 as a default value for user information. The PT values for network control purposes are not yet defined. The Cell Loss Priority (CLP) bit indicates whether a cell has a higher priority (CLP = 0) or is subject to discarding in the network (CLP = 1) in case of congestion. The Reserved (RES) bit allows further enhancement of the cell header for later use. The default value is 0. Finally, the Header Error Control (HEC) field consists of 8 bits.

At the Network Node Interface (NNI) the header format is quite comparable with that at the UNI (Figure 1.7). The only difference lies in the GFC field of the

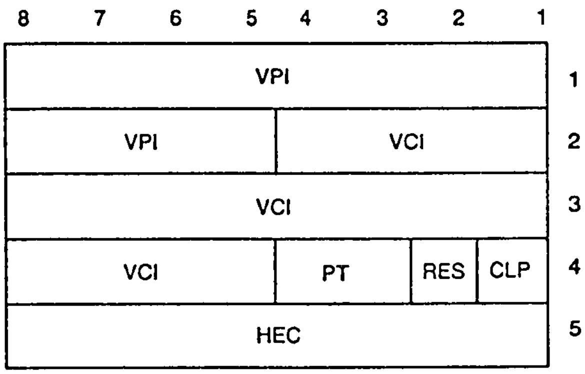

Figure 1.7: ATM header structure at NNI.

UNI which is replaced by 4 additional VPI bits. This results in a VPI field of 12 bits at the NNI.

1.4 Performance Metrics

In order to compare the performance of different techniques, some performance metrics are evidently required. Some metrics are also required to measure the suitability of a network for different types of traffic. Performance metrics for communication networks can be divided into two levels.

- *User-level*, and
- *Network-level.*

At the User-level, the metrics provide information about the quality of service provided by the network resources. This is helpful in determining the type of sources for which a network is suitable. Such metrics are also useful to estimate the state of the network at some point in time. The commonly used metrics at the User level are: *Cell Delay*, and *Cell Loss* .

At the Network-level, performance metrics are used to compare different traffic related mechanisms when the underlying hardware resources are the same. The traffic related mechanisms mainly include the congestion control mechanism and the traffic routing algorithm. The performance metrics at the Network level include *Efficiency, Fairness, Distributedness,* and *Convergence.*

1.4.1 User-level Performance Metrics

Both *Cell delay* and *Cell loss* can be further subdivided into different sub-metrics to evaluate the performance of a network from different angles. Brief definitions of these sub-metrics at the user-level are given below.

Cell Delay (δ)

It is the time interval between the end of transmission of a cell from the source at time t_s and the end of reception of the same cell at the destination at time t_d.

$$\delta = t_d - t_s \tag{1.1}$$

In conventional data packet networks, delay usually occurs due to the low transmission capacity of the transmission channel connecting the nodes of the network. But in ATM networks the propagation delay of the medium and the processing speed of the nodes also make up a substantial part of the overall cell delay between two nodes. This performance sub-metric is important for delay-sensitive applications like voice and video.

Average Cell Delay (d) is the sum of the transfer delays of all the cells divided by the number of cells. If the total number of cells generated in the duration of the connection is N, then

$$d = \frac{\sum_{i=1}^{N} \delta_i}{\sum_{i=1}^{N} i} \tag{1.2}$$

Delay Jitter (J) is the difference between the time of arrivals of two consecutive cells at the destination.

$$J = \max_{1 \le i \le N} (t_{d_i} - t_{d_{i-1}}) \tag{1.3}$$

Cell Loss

It is the number of cells which are not received correctly at the destination. This includes the cells lost during transmission and the cells received with unrecoverable loss to the header. The latter type of cell loss occurs due to transmission errors which are very rare in ATM networks because of the usage of optical fibers as transmission media. Optical fibers have a typical error rate of 10^{-12} to 10^{-15}. This performance parameter is important for loss-sensitive applications like file and data transfer.

Cell Loss Probability (P_{LOSS}) is the probability of not receiving a cell at the destination. This includes receiving cells with the header information damaged to an extent that it can not be recovered by forward error correction.

1.4.2 Network-level Performance Metrics

The performance metrics at the network-level are not standardized. But the following four metrics or criteria [10] are enough to estimate the performance of a congestion control scheme at the network level. Formal definitions of these criteria are as follows:

Efficiency

The *efficiency* of a network resource is defined as the amount of time the resource is used for useful purpose. With reference to Figure 1.8, the efficiency (throughput) of the network is considered good as long as it is between the *knee* and the *cliff*.

It is directly dependent on the bandwidth and the error rate of the transmission medium. In ATM networks, where the available bandwidth is very large and the error rate is very low as compared to the existing packet switching networks, throughput is usually very high.

Fairness

The division of network resources among the users according to their allocated limits is defined as *fairness*. When priorities are used in the allocation of resources, the *maxmin fairness* criterion has been widely adopted. The *maxmin* criterion asserts

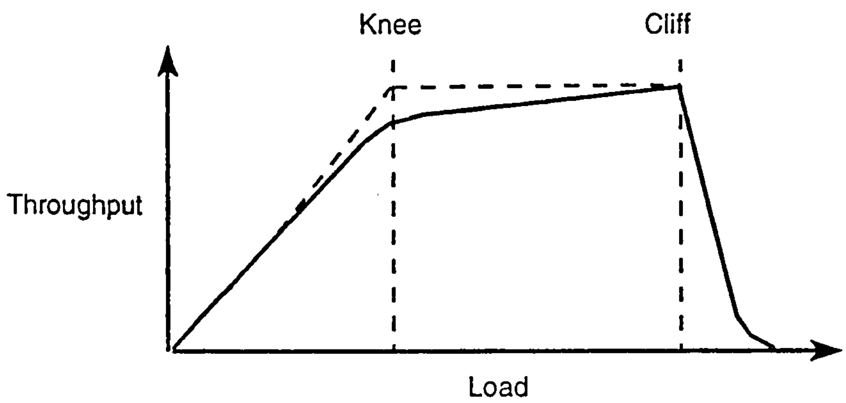

Figure 1.8: Network performance as a function of the load.

that the users in the same equivalent class should have equal share of the contented resource. The *round-robin* manner of resource allocation is proved to be totally fair for equal priority users.

Distributedness

A centralized scheme requires complete knowledge of the state of the system. That is why, in ATM networks, centralized control is considered a major bottleneck for fast reaction to congestion state of the networks. *Distributedness* is the criterion which measures the de-centralization of the congestion control mechanism in the network.

Convergence

Since traffic control is a *control* mechanism, it should converge to an optimal point. Convergence is generally measured by the speed with which this optimal point is reached from any starting state. This criterion is important to guarantee that the mechanism will keep the system from getting unstable.

In binary feedback (which is also used in this thesis), the system does not generally converge to a single steady state. Rather, the system reaches an "equilibrium" in which it oscillates around the optimal state. *Responsiveness* is a sub-criterion of convergence which determines the time to reach the equilibrium state. *Smoothness* is another sub-criterion which determines the magnitude of the oscillations about the optimal point.

Another important parameter which, though not used as a performance metric, is the *Delay-bandwidth product* (Γ). As the name elaborates, it is the product of the transmission bandwidth and the overall delay of the network. This parameter is particularly important in the choice of the traffic control mechanism for a network. If C is the bandwidth (capacity) of the transmission channel, then

$$\Gamma = d \times C \tag{1.4}$$

1.5 Classification of Traffic Control Schemes

Traffic control schemes in computer networks can be classified according to two main design principles:

1. *Reactive control* schemes, and

2. *Preventive control* schemes.

Reactive control schemes are those in which the congested node takes some specified action when it reaches a congested state. Usually this type of control is strongly linked with the flow control mechanism of the network. The congested node sends some feedback to the traffic sources to decrease the traffic rate in the direction of the congested node. This can be performed using any of the flow control schemes. Such congestion control schemes are usually implemented in networks where the delay-bandwidth product is not large. With reference to Figure 1.8, these schemes try to keep the throughput of the network between the knee and the cliff.

Preventive control schemes are those in which the node is not allowed to reach the state of congestion by imposing some strict flow control measures at the entry points to the network. That is why this type of control is also called *congestion avoidance*. While such schemes are good in avoiding congestion at the traffic sources, they are not that effective in preventing congestion at the intermediate nodes in the network. Traffic control schemes based on preventive control are favoured in networks where the delay-bandwidth product is high. Preventive schemes try to keep the throughput of the network around the Knee level (Figure 1.8).

1.6 Traffic Control in Conventional WANs

Congestion control in traditional data networks is a well studied problem. A large number of mechanisms have been developed over the years to solve the problem of congestion control in wide area networks. These mechanisms have been implemented in a number of networks depending on the type of the network and the type of traffic on the network. Next, we survey several of these mechanisms [11]:

Preallocation of Buffers

This congestion control scheme is used in networks where virtual circuit type of connections are setup between the nodes. In this scheme, each virtual connection is allocated a fixed buffer space at each node upon connection establishment. If a connection generates packets greater then its quota the packets are automatically dropped. This scheme is unsuitable for networks with bursty traffic sources because allocating each virtual circuit a fixed number of buffer space may result in wastage of resources.

Flow-Control based Congestion Control

Congestion control schemes based on link level flow control prevent one node from saturating its neighbors. But it is difficult to control the congestion at the intermediate nodes by using end-to-end flow control rules. Still, if the hosts are forced to stop transmitting due to strict flow control rules, the subnet will not be as heavily loaded.

Choke Packets

In this scheme each node monitors the percentage utilization of each of its output lines. At the arrival of each new packet, its output line is checked for being in the warning state. If the output line is found to be in that state, a special packet, called *choke packet*, is send to the source host. The source node, on receiving the choke packet, reduces the traffic to the specified destination.

Timeout-Based Congestion Control

The timeout-based control schemes are based on the idea that packet loss is a good indicator of congestion and therefore, on a timeout, the load on the network should be reduced. Later, if there is no further loss, the load, is increased slowly.

All the congestion control schemes described above (except the first one) are based on the reactive control principle. Because the delay-bandwidth product in conventional data networks is not very large, these schemes give effective for such networks. But reactive control schemes are not very well suited for ATM networks where the delay-bandwidth product is large. Therefore, preventive control schemes are the preferred choice for ATM networks. Most of the traffic control schemes suggested in the literature follow a preventive strategy. These schemes are described in chapter 3. However, preventive control results in a wastage of network resources which cancels the advantage of employing statistical multiplexing in ATM networks.

1.7 Traffic Control Requirements in ATM Networks

There are two main requirements for a traffic control scheme in ATM networks. These are:

- *short processing time (low overhead)*, and
- *good bandwidth utilization.*

The most stringent requirement on a traffic control scheme in ATM networks is the *short processing time*. This is mainly due to the fact that ATM networks also carry real-time traffic along with the conventional non-real time data traffic. Because of the small header, small cell size, and large transmission speed, the detection and response to congestion has to be very swift. If the processing time of the traffic control procedure is long, the congestion might get worse during the processing time of these procedures. This may result in greater cell loss which is not desirable for most traffic sources.

Since the delay-bandwidth product in ATM networks is very large, *end-to-end* reactive traffic control procedures result in a large cell loss before the control measures are taken by the nodes. As most of the traffic sources are bursty in nature, it might happen that by the time a reactive measure comes into effect, the congestion is already over. This will result in a counter effect on the performance of the bandwidth utilization.

Preventive traffic control measures seem more appropriate in such a situation. But the fundamental approach of such methods is to keep the traffic coming into the network much lower than the allowable capacity of the network. This is in conflict with the main concept of statistical multiplexing which is one of the main features of ATM. Therefore, purely preventive traffic control methods can not guarantee adequate level of resource utilization in ATM networks.

As mentioned above, ATM networks employ statistical multiplexing to efficiently use the bandwidth among the different connections. During congestion, the amount of traffic on these connections might be reduced. Since congestion usually occurs due to the occasional bursts from traffic sources, the normal level of traffic can be resumed on the connections when the burst(s) causing congestion is over. The traffic control scheme, therefore, should also be able to restore the normal bandwidth allocation for the connections as quickly as possible for efficient utilization of the available bandwidth.

For an efficient traffic control, ATM networks, therefore, require a scheme which controls the traffic in the networks based on the state of the network, that is, it reacts to the traffic conditions inside the network. But the reaction time should be short enough so that it does not worsen the congestion state. In this thesis, a traffic control scheme fulfilling these requirements has been designed and tested using computer simulation.

Chapter 2

Traffic Source Models

ATM networks are expected to support a wide variety of traffic types. In contrast to conventional data networks, which carry only computer data traffic, ATM networks will carry traffic from multimedia sources which include voice and video as well as computer data. Each of these types of traffic has its own requirements which can be described by certain traffic parameters. Based on these traffic description parameters, QoS negotiations are carried out at the establishment of a connection on an ATM network.

The characterization of ATM traffic sources has been a very active area of research. Traffic source models are important during the negotiation of QoS. They are also essential for admission control. Furthermore, they are needed during simulation studies to predict the cell generation times. In this chapter, first, a brief description of some traffic description parameters is given. Then, the issue of traffic modeling is described in the light of some commonly used traffic models.

2.1 Traffic Description Parameters

A set of traffic descriptors given by a user to a network should include sufficient parameters so that the network can determine the user's traffic characteristics with reasonable accuracy. However, for the sake of simplicity a set of traffic descriptors should include the fewest possible parameters. They should also be easy to determine at the time of establishment of a connection [12].

The most commonly used traffic description parameters are *the peak bit rate, the average bit rate, the maximum and the average burst length,* and *burstiness.* A brief definition of the first four parameters follows with some discussion about the ease and accuracy with which they can be determined at the beginning of the establishment of a connection. With reference to ATM networks, "burstiness" is the most important parameter since congestion is caused by those sources that are highly bursty. Therefore, it will be discussed separately in the next section after a brief introduction of the mathematical terms required for this purpose.

- **Peak Cell Rate** (p): It is the maximum possible rate at which a particular traffic source can generate the cells. For real time traffic sources, like voice and video, it is an easily available parameter at the beginning of a connection. But for conventional computer data traffic, it can only be approximated.

- **Average Cell Rate (m):** It is the number of cells generated over the entire duration of the connection. This parameter is not easy to determine accurately at the time of establishment of a connection. But can be approximated depending on the type of connection (e.g. full motion video, videoconferencing, etc).

- **Maximum Burst Length (L):** This parameter gives the maximum number of cells that can be generated by the source continuously. It provides an estimate of the maximum buffer space requirements for the connection. Again, for real time traffic sources, this parameter can be determined easily at the time of establishment of a connection. But the same does not hold for data traffic.

- **Average Burst Length (N):** This is the average length of all the bursts taken over the entire duration of the connection. As in the case of average bit rate, this parameter is also hard to determine accurately at the beginning of a connection.

2.2 Traffic Characterization

Voice, video and data were the three major types of traffic sources used in communication networks. In the past, separate networks were used for each type of traffic source, i.e. telephone, television and computer respectively. Each type of traffic sources was modelled using different mathematical models and the networks were usually designed to satisfy the requirements of the underlying traffic source type.

As the paradigm of transmission shifted from analog to digital, the need of combining the different networks into one ubiquitous network capable of carrying all types of traffic became realizable. ATM networks is a practical realization of this broadband network. But the modelling requirements of such a network needed a new classification and new models for the traffic sources. For the purpose of classification of traffic sources for ATM networks, the International Telecommunications Union (ITU-T) has categorized the traffic sources into four types:

- Constant Bit Rate (CBR)
- Variable Bit Rate (VBR)
- Unspecified Bit Rate (UBR)
- Available Bit Rate (ABR)

2.2.1 CBR Traffic Type

This type is used to represent the traffic sources which generate cells (or packets) at a constant rate with respect to time. Voice and video sources fall into this category of traffic sources. The mean and the peak cell rates are equal for this type of traffic sources.

$$m_{CBR} = p_{CBR} \tag{2.1}$$

But recent developments in VLSI switching speeds and data compression algorithms have transformed these sources into variable bit rate sources resulting in even better bandwidth utilization.

2.2.2 VBR Traffic Type

VBR traffic sources are those which do not have a fixed cell transmission rate with respect to time. As mentioned above, most CBR traffic can now be converted into VBR type by employing data compression techniques for efficient link bandwidth utilization. This type of traffic sources can be characterized by a mean cell rate and a peak cell rate. Most real time applications, such as videoconferencing etc., are now modelled as VBR sources with specific mean and peak cell rates.

2.2.3 UBR Traffic Type

UBR traffic type is used to specify those traffic sources whose data is not time-critical. The cells for this type of traffic source can be sent across the network without negotiating a fixed bandwidth. If the transporting ATM network uses priority of cells to control congestion, UBR traffic cells are the first to be discarded when congestion begins to happen at a node. Traffic from most conventional computer applications, such as file transfer, e-mail, etc., is of this type.

2.2.4 ABR Traffic Type

This traffic source type is defined by ITU specifically to handle traffic which is unpredictable like conventional data traffic [13]. ABR traffic sources use the ATM network only when no other application (CBR or VBR) is using the network. In a way, they can be visualized as the least priority traffic in an ATM network after CBR and VBR traffic respectively.

2.3 Traffic Modelling

Simple Traffic consists of single arrivals of discrete entities (cells). It can be mathematically described as a point process, consisting of a sequence of arrival instants $T_1, T_2, \ldots, T_n \ldots$ measured from the origin 0; by convention, T_0=0. There are two additional equivalent descriptions of point processes: counting processes and interarrival time processes. A counting process $\{N(t)\}_{t=0}^{\infty}$ is a continuous-time, non-negative integer-valued stochastic process,

$$N(t) = max\{n : T_n \leq t\} \tag{2.2}$$

where $N(t)$ represents the number of (traffic) arrivals in the interval (0,t]. An interarrival time process is a non-negative random sequence $\{A_n\}_{t=0}^{\infty}$, where

$$A_n = T_n - T_{n-1} \tag{2.3}$$

is the length of the time interval separating the nth arrival from the previous one. The equivalence of these descriptions follows the equality of events:

$$\{N(t) = n\} = \{T_n \leq t < T_{n+1}\} = \{\sum_{k=1}^{n} A_k \leq t < \sum_{k=1}^{n+1} A_k\} \tag{2.4}$$

since $T_n = \sum_{k=1}^{n} A_k$. Unless otherwise stated, we assume throughout that $\{A_n\}$ is a stationary sequence and that the common variance of A_n is finite.

Compound Traffic consists of batch arrivals, that is, arrivals may consist of more than one unit at an arrival instant T_n. To fully describe compound traffic, one also needs to specify a non-negative random sequence $\{B_n\}_{t=0}^{\infty}$, where B_n is the number of units in the batch. At a higher level of abstraction, B_n may represent some general attributes of the nth arrival, such as the amount of "work" associated with the nth arrival or its path in a network. Such compound traffic processes are called marked point processes.

Discrete-time traffic processes correspond to the case when time is slotted. Mathematically, this means that the random variables A_n can assume only integer values, or equivalently, that the random variables $N(t)$ are allowed to increase only at integer valued time instants T_n.

Traffic processes are used to drive simulations in several ways, all of which use one or more pseudo-random number streams to generate sequences of random variables via appropriate transformations.

The traffic-generation mechanism that would be contained in the event algorithms is straightforward. Initially, the simulation clock is set to T_0=0. Next, A_1 is randomly generated and an arrival event is scheduled for time $T_1 = A_1$. This arrival event is placed on the chronologically-ordered event list or calender. Eventually, that arrival event will become the most imminent one, the simulation clock will be set to A_1, and that arrival event will be processed. Arrival generation proceeds inductively. At simulation time T_n, the nth arrival event is processed, the next interarrival time A_{n+1} is randomly generated, and an arrival event is scheduled for simulation time $T_{n+1} = T_n + A_{n+1}$ and so on. The processing of a particular event may lead to the scheduling of other types of events. For compound traffic, the simulation randomly generates a batch size B_n (in addition to the interarrival time A_n), and implements the arrival of B_n units at simulation time T_n. Most models call for the sequences $\{A_n\}$ and $\{B_n\}$ to be stochastically independent.

In addition to arrival times and batch sizes, it is often useful to incorporate the notion of workload into the traffic description. The workload is a general concept describing the amount of work $\{W_n\}$ brought to a system by the nth arriving unit; it is usually assumed independent of interarrival times and batch sizes. A typical example is the sequence of service time requirements of arrivals at a queuing system, although in queuing, one usually refers to the arrival process alone as traffic. On the other hand, traffic reduces to workload description when interarrival times are deterministic. A case in point is compressed video, also known as VBR video, where coded frames (arrivals) have variable and random size (bit rate), and these must be delivered deterministically every 1/30 of a second for high-quality video. The workload consists of coded frame sizes (in bits), because frame size is roughly proportional to its transmission time.

In this chapter, generic models that can be used to randomly generate any component of traffic description, be it $\{A_n\}$, $\{B_n\}$, or $\{W_n\}$, will be described but emphasis will be given to simple traffic, described by $\{A_n\}$. It should also be noted that different traffic streams, corresponding to different telecommunications services (voice, video, file transfer, etc.) can be multiplexed to form a realistic heterogeneous mixture of traffic.

2.3.1 Traffic Burstiness (β)

A recurrent theme relating to traffic in broadband networks is the traffic burstiness exhibited by key services such as compressed video, file transfer, etc. Burstiness is present in a traffic process if the arrival points $\{T_n\}$ appear to form visual clusters, that is, $\{A_n\}$ tends to give rise to runs of several relatively short interarrival times followed by a relatively long one. The mathematical representation of burstiness is more complex. Two main sources of burstiness are due to the shapes of the marginal distribution and autocorrelation function of $\{A_n\}$. For example, burstiness would be facilitated be a bimodal marginal distribution of $\{A_n\}$, or by short-term autocorrelations in $\{A_n\}$. Strong positive autocorrelations are a particularly major cause of burstiness. Since there seems to be no single widely-accepted notion of burstiness, a brief description of some of the commonly-used mathematical measures that attempt to capture it is given here.

The two simplest measures of burstiness take account only of first-order properties of traffic. They are each a function of the marginal distribution only of interarrival times. The first one is the ratio of the peak rate to mean rate. This is a very crude measure, which also has the shortcoming of dependence on the interval length utilized for rate measurement. A more elaborate measure of burstiness is the coefficient of variation, defined as the ratio of standard deviation to mean of interarrival intervals.

$$\beta = \frac{\sigma[A_n]}{E[A_n]} \tag{2.5}$$

In contrast, the peakedness measure [14] and the index-of-dispersion measure [15] do take into account the temporal dependence in traffic (second-order properties). For a given time interval of length τ, the index of dispersion for counts (IDC) is the function

$$I_c(\tau) = \frac{Var[N(\tau)]}{E[N(\tau)]} \tag{2.6}$$

i.e., the variance-to-mean ratio of the number of arrivals in the interval $[0,\tau]$. Since the number of arrivals is related to the sum of interarrival intervals via Equation 2.4, the numerator of the IDC includes the autocorrelations of $\{A_n\}$. The peakedness concept is similar, but more involved. Assume that the traffic stream $\{A_n\}$ is offered to an infinite server group consisting of independent servers with common service time distribution F. Let S be the number of busy servers at steady state. The peakedness is the function

$$z_A[F] = \frac{Var[S]}{E[S]} \tag{2.7}$$

which maps a service time distribution to a real number. A commonly used peakedness is $z_{exp}(0)$, obtained as a limiting case for an exponential service distribution with service rate approaching 0.

2.4 Renewal Traffic Models

This section introduces renewal traffic processes and the important special cases of Poisson and Bernoulli processes. Renewal models have a long history, because of their relative mathematical simplicity. In a renewal traffic process, the A_n are independent & identically distributed (IID), but their distribution is allowed to be general. Unfortunately, with few exceptions, the superposition of independent renewal processes does not yield a renewal process. The ones that do, however, occupy a special position in traffic theory and practice. Queuing models historically have routinely assumed a renewal-offered traffic.

Renewal processes, while simple analytically, have a severe modeling drawback – the autocorrelation function of $\{A_n\}$ vanishes identically for all nonzero lags. The importance of capturing autocorrelations stems from the role of the autocorrelation functions as a statistical proxy for temporal dependence in time series. Moreover, recall that positive autocorrelation in $\{A_n\}$ can explain, to a large extent, the phenomenon of traffic burstiness. Bursty traffic is expected to dominate broadband networks, and when offered to a queuing system, it gives rise to much worse performance (such as mean waiting times) as compared to renewal traffic (which lacks temporal dependence). Consequently, models that capture the autocorrelated nature of traffic are essential for predicting the performance of emerging broadband networks.

2.4.1 Poisson Processes

Poisson models are the oldest traffic models, dating back to the advent of telephony and the renowned pioneering telephone engineer A. K. Erlang. A Poisson process can be characterized as a renewal process whose interarrival times $\{A_n\}$ are exponentially distributed with rate parameter λ:

$$P\{A_n \leq t\} = 1 - e^{-\lambda t} \tag{2.8}$$

Equivalently, it is a counting process, satisfying

$$P\{N(t) = n\} = \frac{(\lambda t)^n e^{-\lambda t}}{n!} \tag{2.9}$$

and the number of arrivals in disjoint intervals is statistically independent.

Poisson processes enjoy some elegant analytical properties. First, the superposition of independent Poisson processes results in a new Poisson process whose rate is the sum of the component rates. Second, the independent increment property renders Poisson a memoryless process. This, in turn, greatly simplifies queuing problems involving Poisson arrivals. Third, Poisson processes are fairly common in traffic applications that physically comprise a large number of independent traffic streams, each of which may be quite general. The theoretical basis for this phenomenon is known as Palm's Theorem. It states that under suitable but mild regularity conditions, such multiplexed streams approach a Poisson process as the number of streams grows, but the individual rates decrease so as to keep the aggregate rate constant. Thus, traffic stream on a main communication channel is commonly believed to follow a Poisson process, as opposed to traffic on its constituent upstream channels, which are less likely to be Poisson.

2.4.2 Bernoulli Processes

Bernoulli processes are the discrete-time analog of Poisson processes. Here the probability of an arrival in any time slot is p, independent of any other one. It follows that for k consecutive slots, the corresponding number of arrivals is binomial,

$$P\{N_k = n\} = \binom{k}{n} p^n (1-p)^{k-n}, \qquad 0 \leq n \leq k \tag{2.10}$$

The time between arrivals is geometric with parameter p,

$$P\{A_n = j\} = p(1-p)^j \tag{2.11}$$

j being a non-negative integer and A_n is defined in Equation 2.3.

2.5 Markov and Markov-Renewal Traffic Models

Unlike renewal traffic models, Markov and Markov-renewal traffic models introduce dependence into the random sequence $\{A_n\}$. Consequently, they can potentially capture traffic burstiness, because of nonzero autocorrelations in $\{A_n\}$.

Consider a continuous-time Markov process M=$\{M(t)\}_{t=0}^{\infty}$ with a discrete state space. In this case, M behaves as follows: it stays in a state i for an exponentially distributed holding time with parameter λ_i, which depends on i alone; it then jumps to state j with probability p_{ij}, such that the matrix $P = [p_{ij}]$ is a probability matrix. In a simple Markov traffic model, each jump of the Markov process is interpreted as signaling an arrival, so interarrival times are exponential, and their rate parameters depend on the state from which the jump occurred. This results in dependence among successive interarrival times as a consequence of the Markov property.

Markov models in slotted time can be defined for the process $\{A_n\}$ in terms of a Markov transition matrix $P = [p_{ij}]$. Here, state i corresponds to i idle slots separating successive arrivals, and p_{ij} is the probability of having a j-slot separation, given that the previous one was an i-slot separation. Arrivals may be single units, a batch of units, or a continuous quantity. Batches may themselves be described by a Markov chain, whereas continuous-state, discrete-time Markov processes can model the workload arriving synchronously at the system. In all cases, the Markov property introduces dependence into interarrival separation, batch sizes and successive workloads, respectively.

Markov-renewal models are more general than discrete-state Markov processes, yet retain a measure of simplicity and analytical tractability. A Markov renewal process $R = \{(M_n, \tau_n)\}_{n=0}^{\infty}$ is defined by a Markov chain $\{M_n\}$ and its associated jump times $\{\tau_n\}$, subject to the following constraint: the pair (M_{n+1}, τ_{n+1}) of next state and inter-jump time depends only on the current state M_n, but not on previous states nor on previous inter-jump times. Again, if we interpret jumps (transitions) of $\{M_n\}$ as signaling arrivals, we would have dependence in the arrival process. Also, unlike the Markov process case, the interarrival times can be arbitrarily distributed, and these distributions depend on both states straddling each interarrival interval.

2.6 Markov-Modulated Traffic Models

Markov-modulated models constitute an extremely important class of traffic models. The idea is to introduce an explicit notion of state into the description of a traffic stream – an auxiliary Markov process is evolving in time and its current state controls (modulates) the probability law of the traffic mechanism.

Let $M = \{M(t)\}_{t=0}^{\infty}$ be a continuous-time Markov process, with state space $\{1, 2, \ldots, m\}$. Now assume that while M is in state k, the probability law of traffic arrivals is completely determined by k, and this holds for every $1 \leq k \leq m$. Note that when M undergoes a transition to, say, state j, then a new probability law for arrivals takes effect for the duration of state j, and so on. Thus, the probability law for arrivals is modulated by the state of M (such systems are also called double stochastic).

The modulating process certainly can be more complicated than a continuous-time, discrete-state Markov process (so the holding times need not be restricted to exponential random variables), but such models are far less analytically tractable. For example, Markov Renewal-modulated processes constitute a natural generalization of Markov-modulated processes with generally-distributed interarrival times.

2.6.1 Markov-Modulated Poisson Processes

The most commonly used Markov-modulated model is the Markov-Modulated Poisson Process (MMPP) model, which combines the simplicity of the modulating (Markov) process with that of the modulated (Poisson) process. In this case, the modulation mechanism simply stipulates that in state k of M, arrivals occur according to a Poisson process at a rate λ_k. As the state changes, so does the rate.

MMPP models can be used in a number of ways. Consider first a single traffic source with a variable rate. A simple traffic model would quantize the rate into a finite number of rates, and each rate would give rise to a state in some Markov modulating process. It remains to verify that exponential holding times of rates are an appropriate description, but the Markov probability transition matrix $Q = [q_{kj}]$ of the supposed M can be easily estimated from empirical data, and then estimate

q_{kj} by calculating the fraction of times that M switches from state k to state j.

As a simple example, consider a two-state MMPP model, where one state is an "ON" state with an positive Poisson rate, and the other is an "OFF" state with associated rate zero[1]. This ON-OFF model has been widely used to model voice traffic sources [16]. This basic MMPP model can be extended to aggregations of independent traffic sources, each of which is an MMPP, modulated by an individual Markov process M_i, as described previously. Let

$$J(t) = (J_1(t), J_2(t), \ldots, J_r(t)), \tag{2.12}$$

where $J_i(t)$ is the number of active sources of traffic type i, and let

$$M(t) = (M_1(t), M_2(t), \ldots, M_r(t)) \tag{2.13}$$

be the corresponding vector-valued Markov process taking values on all r-dimensional vectors with non-negative integer components. The arrival rate of class i traffic in state $(j_1, j_2, \ldots, j_i, \ldots, j_r)$ of $M(t)$ is $j_i\ \lambda_{it}, \quad 1 \leq i \leq r$.

2.7 Fluid-flow Traffic Models

The fluid traffic paradigm dispenses with individual traffic units. Instead, it views traffic as a fluid stream, characterized by a flow rate (such as bits per second), so that a traffic count is replaced by a traffic volume.

[1]Such models are also known as interrupted Poisson

Fluid models are appropriate to cases where individual units are numerous relative to a chosen time scale. In other words, an individual unit is by itself of little significance. In the context of ATM, all packets are fixed-size cells of relatively short length (53 octets). In addition, the high transmission speeds render the transmission impact of an individual cell negligible. This can be observed if we contrast an ATM cell with a much higher transmission unit, such as a compressed high-quality video frame, which may consist of over a thousand cells. A traffic arrival stream of such frames should be modeled as a discrete stream of arrivals, because such frames are typically transmitted at the rate of 30 frames per second. A fluid model, however, is appropriate for the constituent cells.

Although an important advantage of fluid models is their conceptual simplicity, important benefits will also accrue to a simulation model of fluid traffic. For example, consider the ATM scenario. If one is to distinguish among cells, then each of them would result in several events. The time granularity of event processing would be quite fine, and consequently, processing cell arrivals would consume large CPU and memory resources, even on simulated time scales of minutes. A statistically meaningful simulation may often be infeasible. In contrast, a fluid simulation would assume that the incoming fluid flow remains constant over much longer time periods. Traffic fluctuations are modeled by events signaling a change of flow rate. Because these changes can be assumed to happen far less frequently than individual cell arrivals, one can realize enormous savings in computing. In fact, infeasible simulations of cell arrival models can be replaced by feasible simulations of fluid models of comparable accuracy. In a queuing context, it is easy to manipulate fluid buffers. Furthermore, the waiting time concept simply becomes the time it takes to serve the current buffer, and loss probabilities can be calculated in terms of overflow volumes. Because fluid models assume a deterministic service rate, these statistics can be readily computed. Typically though, larger traffic units (such as frames) are of greater interest than individual cells. Modeling the large units as discrete traffic and their transport as fluid flow would give us the best of both worlds: we can measure waiting times and loss probabilities and save on simulation computing resources.

Typical fluid models assume that sources are bursty, i.e., they are of the "ON-OFF"type. While in the OFF state, traffic is switched off, whereas in the ON state traffic arrives deterministically at a constant rate λ. For analytical tractability, the duration of the ON and OFF periods are assumed to be exponentially distributed and mutually independent, i.e., they form an alternating renewal process. Fluid traffic models of these types can be analyzed as Markov-modulated constant rate traffic. The host of generalizations, described above for MMPP, carries over to fluid models as well, including multiple sources and multiple classes of sources.

2.8 Autoregressive Traffic Models

Autoregressive models define the next random variable in the sequence as an explicit function of previous ones within a time window stretching from the present into the past. Such models are particularly suitable for modeling VBR-coded video. The nature of video frames is such that successive frames within a video scene vary visually very little. Only scene changes and other visual discontinuities can cause abrupt changes in frame bit rate. Thus, the sequence of bit rates comprising a video scene may by modeled by an autoregressive scheme, while scene changes can be modeled by some modulating mechanism, such as a Markov chain.

2.8.1 Linear Autoregressive Models

The class of linear autoregressive models has this [17] form:

$$X_n = a_0 + \sum_{r=1}^{p} a_r X_{n-r} + \varepsilon_n \qquad n > 0 \tag{2.14}$$

where $X_0, X_1, \ldots, X_{n-1}$ are prescribed random variables, the a_r are real constants, and the ε_n are zero-mean, IID random variables, called residuals, which are independent of the X_n.

Equation 2.14 describes the simplest form of a linear autoregression scheme, called *AR(p)*, where p is the order of the autoregression. In a good model, the residuals ought to be of a much smaller magnitude than the X_n, in order to explain the empirical data.

X_n in Equation 2.14 can represent any randomly generated parameter of the traffic model which has an autoregressive property. The recursive form in Equation 2.14 makes it clear how to randomly generate the next random element in the sequence $\{X_n\}_{n=0}^{\infty}$ from a previous one. This simplicity makes AR schemes popular candidates for modeling autocorrelated traffic. A simple AR(2) model has been used to model VBR coded video [18]. More elaborate models can be constructed out of $AR(p)$ models combined with other schemes [17].

2.9 Self-Similar Traffic Models

Recent studies of high-quality, high-resolution traffic measurements have revealed a new phenomenon with potentially important ramifications to the modeling, design, and control of broadband networks. These include an analysis of hundreds of millions of observed packets over an Ethernet LAN in a R&D environment [19], and an analysis of a few millions of observed frame data generated by VBR video services [20]. In these studies, packet traffic appears to be statistically self-similar. A self-similar (or fractal) phenomenon exhibits structural similarities across all the time scales. In the case of packet traffic, self-similarity is manifested in the absence of a average length of a burst i.e., at every time scale ranging from a few milliseconds to minutes and hours, similar looking traffic bursts are evident.

Chapter 3

Literature Review

In present day communication networks, most traffic sources are bursty. This is because the number of computers getting attached to the communication networks is increasing more than linearly. Also the compression and coding techniques being adopted transform the previously constant bit rate applications, like voice and video, into variable bit rate applications [21]. An ATM network supports a large number of bursty traffic sources and uses statistical multiplexing to ensure efficient bandwidth utilization. But if a large number of traffic sources become active simultaneously, severe network congestion can occur.

As described in [22], traffic control schemes based on *end-to-end* reactive control mechanism are not very suitable for high speed networks, like ATM, due to the large bandwidth and high transmission speeds. This is also generally shown by most researchers [23-33]. Therefore, most of the proposed congestion control schemes are preventive. The most popular preventive strategies include: access control (admission control) [23, 24, 25, 26, 27, 28, 29, 30, 31], rate control (traffic policing) [31, 32], and transmission control [33, 34, 35]. But some schemes based on reactive control have been proposed and implemented in present day networks [36, 37].

The principle of access control is the regulation of traffic flow into the network with the objective of maintaining the network load below congestion level [23, 27]. The simulation results reported have shown that the use of access control without a proper transmission control can lead to cells being accumulated at the intermediate nodes forming long cell bursts [34]. This is due to the asynchronous nature of the ATM network and the interaction of different cell streams.

Rate control focuses on controlling the rate of each connection separately, by controlling the number of cells from each connection which can be transmitted in a given time. The rate of each connection can be controlled at the network access points and/or at the intermediate nodes along its route. The Leaky Bucket scheme and its variants are among the most popular schemes in this category [38, 31, 32]. However, it has been shown that due to interactions among different connections at intermediate nodes, independent control of connections may be insufficient to guarantee the required QoS for each connection.

Transmission control or transmission scheduling refers to the scheduling of cells from all output queues onto an outgoing link [35, 39]. A transmission control scheme considers all the connections routed through a link and typically decides not only on the number of cells to be transmitted during a given time period from each connection, but also on the order in which these transmissions should occur. Transmission control differs from the rate control in that it coordinates the cell flow of all the connections instead of controlling them independently.

In ATM networks, traffic control can be divided into two main levels; Call (or VC) level and Cell level [38]. Call level traffic control is performed at the source nodes where a connection is originated. Cell level traffic control takes place after the call has been accepted for transmission by the node.

The work in this thesis is focussed on the traffic control at the cell level. However a brief literature review for the call level traffic control research is also described in the following section to differentiate between the two levels of control.

3.1 Call-Level Traffic Control

Since each node has a fixed bandwidth which it can divide among different users (or calls), a new call request is accepted only if the node has the available bandwidth that is requested by the connection request. This is implemented at the node by the Admission Control schemes. Admission control is also considered as part of traffic control by many researchers. DuBose and Kim [21] have proposed a Table Lookup Based admission control scheme in which the *effective bandwidths* of the new connection requests are used to decide whether the connection should be setup or not. Sykas *et al.*, [40] have analyzed a similar scheme for the ON-OFF model of ATM sources. Gallassi *et al.*, [25] have proposed an admission control (bandwidth allocation) scheme based on user characterization according to the classification of the traffic similar to the one used in plain telephony i.e., according to the time of the day. But this scheme is not suitable for networks of global level due to the time differences at different points of the network. Cooper and Park [23] have proposed to preallocate a percentage of available bandwidth into smaller portions for groups of users and the decision to admit a new call is taken at the group level. But this scheme may result in inefficient bandwidth utilization if the groups are not properly selected. Peha [41] has proposed a comprehensive scheme for admission control, scheduling, and policing in ATM networks. His scheme makes use of *priority* assignment to achieve this objective.

Li [42] has proposed an admission control scheme for connectionless traffic on ATM networks. The basis of his scheme is a bandwidth computation technique that establishes a direct association between the number of users of a connectionless application and the minimum amount of bandwidth required under the constraint of congestion control. Abe and Soumiya [43] have suggested two traffic control schemes separately for call-level and cell-level traffic control. The call-level scheme is based on the statistical prediction of traffic flow of a call based on the call traffic parameters by using Gaussian and Poisson distributions. Chatterjee and Bassiouni [44] have also approached the traffic control problem in a two-level approach. They have used the fluid-flow approximation for calculating the equivalent bandwidth of a call using the peak rate, activity ratio and burst period as parameters for the model.

Jeon and Viniotis [45] have proposed a traffic control scheme based on conservation laws. They have derived a conservation law for cell loss probabilities, for an ATM node with a finite buffer size and correlated arrival processes. Based on this law, they have developed a buffer allocation algorithm that meets a prespecified cell loss objective. A novel call admission control scheme based on the theory of co-operative games has been suggested by Mason *et al.* [46]. They have computed the *Nash, Raiffa-Kalai-Smorodinsky* and *Modified Thomson* solutions for a multi-channel link supporting multiple call classes which differ by bandwidth and holding time requirements. The measure of performance in their paper is the call loss probability.

The performance of three types of admission policies is studied by Raha *et al.* [47]. The admission policies they have studied are the *First Come First Serve, Round Robin* and *Packet-by-Packet Generalized Processor Sharing.* But the study is done for ATM LANs in which traffic control is not complicated due to the short propagation delay. Nevertheless, for real-time applications even LANs have an upper bound for delay which requires effective admission control policies.

Since call setup time does not involve any fast cell traffic, it does not require a fast response. That is why even neural networks based [48] and learning algorithm based [49] call level congestion control has been proposed in the literature.

3.2 Cell-Level Traffic Control

Cell-Level traffic control schemes are employed to control congestion in the ATM network due to the traffic generated by the sources which have been allowed to establish a connection i.e., after they pass through the call admission control scheme successfully. Such schemes include usage parameter control (also known as traffic policing), which enforces contracts specifying key parameters such as average and peak rates and maximum burst lengths. In addition, some classes of traffic, which are not very delay sensitive can be shaped by buffering. Cell level traffic control schemes can further be sub-divided into two categories:

- Traffic control at the source nodes, and
- Traffic control at all the nodes.

3.2.1 Traffic Control at Source Nodes

Traffic control at the source nodes is mostly carried out by using the "Leaky Bucket" scheme or any of its variants [38]. The leaky bucket traffic control scheme is a popular and widely studied scheme [40, 54-59]. The main reason for its popularity is its flexibility in the use of different parameters for the selection of the token generation rate and its low processing overhead. In a simple leaky bucket scheme, a cell is accepted only when it can draw a token from a token pool (Figure 3.1). The tokens are generated at a rate which is mainly a function of the bandwidth

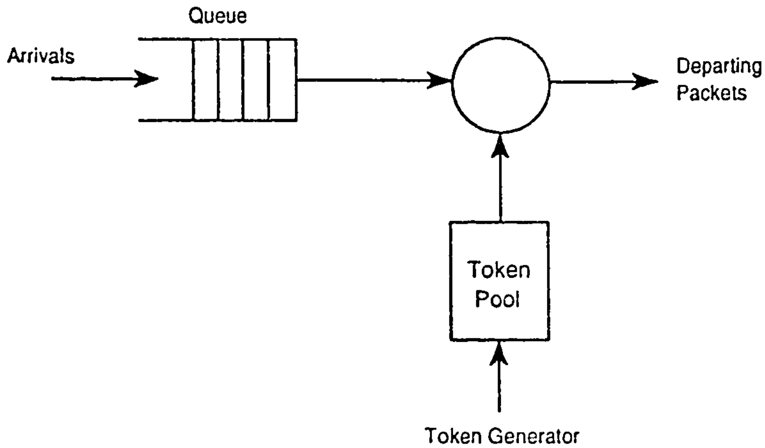

Figure 3.1: A queuing model for a leaky bucket method.

allocated to the particular connection at the connection setup time. Usually the bandwidth allocated is less than the peak rate of the call to improve the efficiency of the statistical multiplexing used in ATM. There might be a situation where the node has the free bandwidth available which can be utilized to accept the extra cells generated by a bursty source in excess of its allocated bandwidth. In order to accommodate such a situation, the extra cells are accepted into the network but are marked by some special tokens (e.g., red tokens as against green for normal cells [38]). Such cells can be dropped at any point in the network in case a congestion is encountered.

Priority shaping of the source traffic is also used as a mean of traffic control at the source nodes in some cases [50]. It is also referred to as traffic shaper function or algorithm. A shaper can be devised to optimize the usage of bandwidth available to the source under its contract with the network to improve the service of the high-priority cells at a cost of delay of the low-priority data cells. This is achieved by implementing a 'ghost' policer as part of the shaper and utilizing a priority based service scheme. This policer utilizes identical parameters as those of the network policer. Reiss [50] has proposed the scheme for traffic streams with two priority levels. Although it can be extended to any number of priority assignments. Bernabei *et al.* [51] have proposed a two level shaper for traffic sources which have been modelled as ON-OFF sources. They have provided an analytical study of the effect of the shaping on congestion control.

Congestion avoidance for ATM networks has also been studied as a mathematical problem in [40]. The difficulty in this approach stems from two facts. Firstly, there has not been enough information available on the traffic characteristics of the real traffic sources that is assumed will be supported by future B-ISDNs. Secondly, there is no agreement on the methods that can be used for the approximation of the superposition of many traffic sources with different and sometimes antagonistic characteristics. Sykas *et al.* [40] have assumed two types of traffic sources they have modelled using the ON-OFF model.

Many researchers have studied the performance of the different rate-based traffic control schemes [52, 53, 54, 55, 56]. Most of these researchers have used different traffic models to study the leaky bucket scheme under their traffic models. Some other rate control mechanisms have also been studied by some researchers [53, 57]. These are the Jumping window, the Triggered Jumping window, the Moving window, the Rectangular Sliding window, the Triangular Sliding window and the Exponentially Weighted Moving Average (EWMA) mechanisms.

Dittmann *et al.* [57] have evaluated four traffic enforcement algorithms. The algorithms are the leaky bucket, the rectangular sliding window, the triangular sliding window, and the exponentially weighted moving average. The basis of their comparison is a mix of teletraffic and signal processing theory. Their analysis shows that the leaky bucket mechanism and EWMA are not only cost-effective but also flexible in implementation. Rathgeb [53] has compared five different mechanisms, namely, the leaky bucket, the jumping window, the triggered jumping window, the moving window, and the exponentially weighted moving average mechanisms. His comparison also shows that the leaky bucket and EWMA are the most suitable traffic control mechanisms for ATM networks.

Buttó *et al.* [54] have performed an analysis of the leaky bucket mechanism under the assumption of "fluid flow" traffic model. They concluded that, while the leaky bucket mechanism can easily control the peak rate of the offered traffic, difficulties arise in controlling the mean bit rate and the burst duration. Sohraby and Sidi [52] have analyzed the performance of the leaky bucket mechanism for the case of bursty and modulated traffic sources. They have modelled the cell arrivals as discrete Markovian arrival process. Their results show that for sources with relatively large active periods the token generation rate should be chosen to be close to the peak rate of the source, and increasing the bucket size of the leaky bucket does not improve the performance substantially. Kim *et al.* [55] have studied the performance of the leaky bucket mechanism under the Markov Modulated Poisson Process (MMPP) source model. Their analysis shows, that under MMPP the trade-off between the cell delay and the cell blocking probability can be controlled by the size of the cells input buffer. They also showed that the performance of the traffic improves with the increase in the size of the token pool.

Abe and Soumiya [43] have also proposed a cell-level congestion control scheme based on buffer management policy, along with the call admission control scheme described above. Their scheme discards cells based on dual threshold level of the buffer. On reaching the first threshold, the tagged cells are discarded. On the second threshold level, priority-based discarding is performed. Hsing [56] has studied the performance of the leaky bucket mechanism with the use of Cell Loss Priority (CLP) bit in the ATM header. This scheme also uses two threshold levels for taking the appropriate traffic control actions. In his scheme the noncompliant cells are marked when a certain buffer threshold level is reached. Some of these marked cells can be dropped with some probability. If the second threshold level is crossed, all the cells beyond that threshold are dropped. The analysis shows that the scheme is good for traffic with small average burst lengths.

Chlamtac and Zhang [39] have proposed a counter based congestion control scheme. Their scheme uses separate counters for different classes of traffic streams which are incremented as the cells are transmitted. Their scheme also performs traffic shaping on the different classes of traffic. But the scheme does not utilize the bandwidth to its maximum and behaves more like time division multiplexing.

A congestion control mechanism based on selective packet discarding has also been suggested [58], though not explicitly for ATM networks. This scheme can be used in ATM network as stated in [22]. But this scheme puts extra demand on the processing power (speed) of the switch since it has to decide which packets to drop so that the QoS is not degraded. On a congested node this is very difficult to implement. Moreover, the time spent in selecting the right packets to drop may be more efficiently utilized to transmit packets and thus avoid congestion.

3.2.2 Traffic Control at Intermediate Nodes

Traffic control at the intermediate nodes is a much more complex issue than at the source nodes. Congestion at the intermediate nodes can occur even when the traffic control schemes are in effect at the source nodes [25]. The main reason for this is that ATM nodes allocate bandwidth to channels on the basis of statistical multiplexing. If some traffic sources, located at different source nodes but whose traffic pass through the same intermediate node, generate traffic bursts greater than their average rate at the same time, congestion occurs at that particular intermediate node.

Two schemes based on preventive control principle have been described for inter-

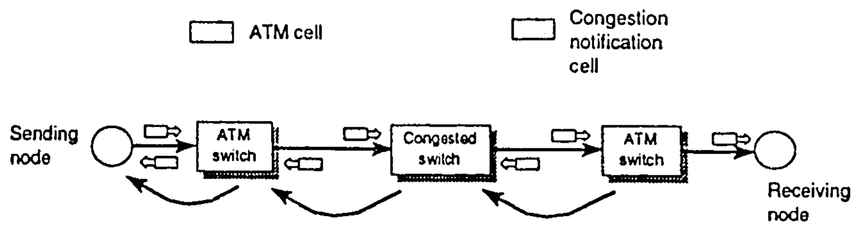

Figure 3.2: Virtual circuit flow control.

node traffic control in the literature [37]. These are known as *Virtual Circuit Flow Control (VCFC)* or Credit-based congestion control. In these schemes, congestion is monitored on links connecting every node in the network. "Credit" messages are exchanged between network nodes to ensure that cells are passed from one node to the next only if there is sufficient buffer space available at the next node (Figure 3.2). This places an extra congestion control burden on ATM switches, in that, every intermediate switch in the network is involved in flow control, and must be equipped with the intelligence to count cells in and out of each buffer.

In one of the schemes, credit messages are passed by specially created cells. The other scheme inserts the credit messages into the headers of cells that are passing through switches, thus saving bandwidth. In terms of performance, credit-based schemes would allow ATM networks to react to congestion fastest, since they monitor it on a link-by-link basis. But they are the most expensive to implement because of the extra buffering and intelligence required at each switch.

Although traffic control in ATM networks based on reactive control have not been found very effective, schemes based on this principle have been proposed and implemented in ATM switches for traffic control at the intermediate nodes. Two schemes are largely popular in this context, namely, Forward Explicit Congestion Notification (FECN) and Backward Explicit Congestion Notification (BECN) [37, 32]. Both the schemes have been successfully implemented in Frame Relay networks. Their ease of implementation and requirement of less hardware makes them attractive. But these schemes are effective only when the applications running on the ATM networks do not require a fast response. At present, the load on ATM networks is not so much to expose the deficiency of FECN and BECN schemes. A brief description of these schemes is given below:

Backward Explicit Congestion Notification *(BECN):*

With BECN, the ATM switches in the network monitor congestion by counting the number of cells held in their incoming buffers. If at a particular switch the buffers start to fill, signifying a potential congestion problem, the switch sends a congestion notification message back to the corresponding source nodes, directing them to throttle back data rates (Figure 3.3).

Forward Explicit Congestion Notification *(FECN):*

FECN works in a slightly different way. While it also monitors congestion by counting buffered cells, when a potential congestion problem exists, the switch inserts a congestion notification bit into the headers of cells, and then sends those cells

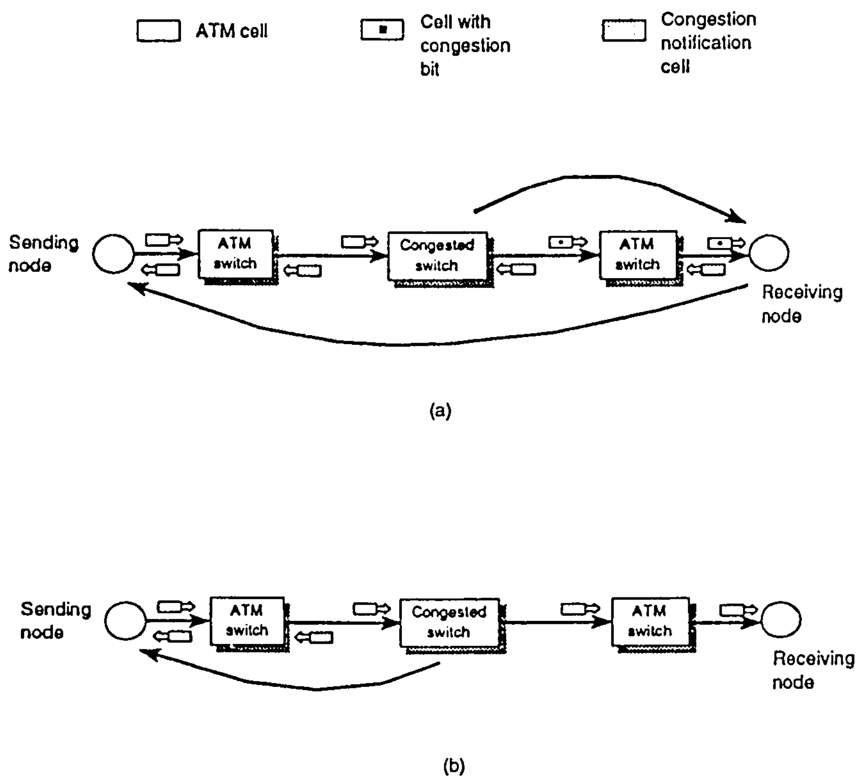

Figure 3.3: (a) FECN (b) BECN.

along to their intended target. At the receiving site, the ATM end node reads the congestion bit and then sends a congestion notification cell back to the originating node, telling it to throttle back (Figure 3.3).

It is clear from the description above that both the FECN and the BECN schemes are based on controlling congestion by performing flow control on the source nodes. The main difference between BECN and FECN is that BECN requires a slightly more active role for the ATM switch, and the congestion notification reaches the sending node faster.

Jain and Ramakrishnan [59] give a good conceptual survey of the issues faced in congestion control of data traffic. Siu and Tzeng [32] have given a congestion control scheme for this type of traffic based on adaptive rate control. Their scheme uses iterative estimation of the optimal cell rate for each VC. This information is then conveyed to the sources with a positive feedback mechanism.

Chapter 4

A Queue Length Based Reactive Traffic Control Scheme

This chapter describes the scheme proposed in this thesis. The motivation for the development of the scheme is explained first. Then a general overview of the scheme is presented. This is followed by an in depth look at the main issues that have been resolved to make the scheme practical for implementation in an ATM networking environment.

4.1 Motivation

The design of the proposed scheme is motivated by two conflicting traffic control requirements in ATM networks. These are the *reactive control* and the *fast response* *time.*

Traditionally, traffic control in networks with high delay-bandwidth product, e.g. those based on satellite links, is performed using the preventive approach. Although ATM networks also fall into the same class of networks, the preventive approach is not suitable for ATM networks. The main reason for this non-suitability is the usage of statistical multiplexing in ATM networks, which is one of its major characteristics which gives it superiority over other communication techniques.

Reactive traffic control has been applied in computer networks where the delay-bandwidth product was low. This type of traffic control is common in point-to-point networks based on heterogeneous types of computers. This approach of traffic control was applied on a node-to-node level as well as on an end-to-end in earlier networks. But as the bit error rate of the transmission media dropped and better coding techniques were developed, node-to-node error control was phased out from the computer networks.

The response time of the reactive end-to-end traffic control depends upon the bandwidths of the transmission media used by the network. The data in transit during the response time is usually lost and has to be retransmitted. For low bandwidth links, the amount of lost data is small. But in ATM networks with high speed links, the response time is more significant and results in voluminous retransmissions. This results in a substantial reduction in the efficiency of the network throughput.

4.1.1 International Telecommunication Union's (ITU) Recommendations

Since traffic control is an important issue in ATM network, it has been adequately addressed by the ITU. Due to the different possible implementation scenarios in an ATM network, ITU has not specified any rigid implementation standard for traffic control. But some important recommendations have been made consisting of some efficient traffic control algorithms. The two most important algorithms are:

- Generic Cell-Rate Algorithm (GCRA), and
- Backward Explicit Congestion Notification (BECN).

4.2 Overview of the Scheme

To overcome the above mentioned deficiencies of the conventional traffic control schemes and to solve the problem under the guidelines of the ITU recommendations, we propose a new traffic control scheme. This new scheme is reactive in essence but has fast response time to minimize the loss of data. Both GCRA (better known as Leaky Bucket) and BECN are used in the scheme to control the congestion at two different levels in the network. The Leaky Bucket control is used at all the sources and BECN-based scheme is used at every transit node of the network. A node is called a *source node* if it has one or more traffic sources attached to it. A node is called a *transit node* if it has the possibility of relaying traffic from upstream nodes to the downstream nodes.

Two types of queues are used in the scheme to store the cells of the traffic sources inside the network[1]. The first type is called the *Source queue.* It is used to store the traffic from a particular source and is situated at the originating node for each source. The second queue type stores all those cells at a node that are generated at upstream nodes and destined to a downstream node. This type of queue is called *Transit queue.* Both queue types follow a *First-In-First-Out* (FIFO) queuing discipline.

The first type of traffic control takes place at the nodes where the traffic sources are attached. The traffic from the sources is controlled by the nodes using the Leaky Bucket algorithm. The second type of traffic control controls the congestion at the transit nodes caused by the traffic coming from upstream nodes. Each transit node monitors its transit queue. Once it detects congestion, it transmits *backpressure* notifications to appropriate immediate predecessor nodes (Figure 4.1). Upon reception of a backpressure notification, a node takes immediate action to reduce its traffic to its successor nodes. But it should be noted that this reduction takes place ONLY for those traffic sources which are transmitting towards the congested node. Other traffic sources attached to the node are not affected at all. Backpressure notification is also sent to upstream nodes once the congestion state is over (Figure 4.2). These

[1]The modelling of these queue types is described in detail in the next chapter.

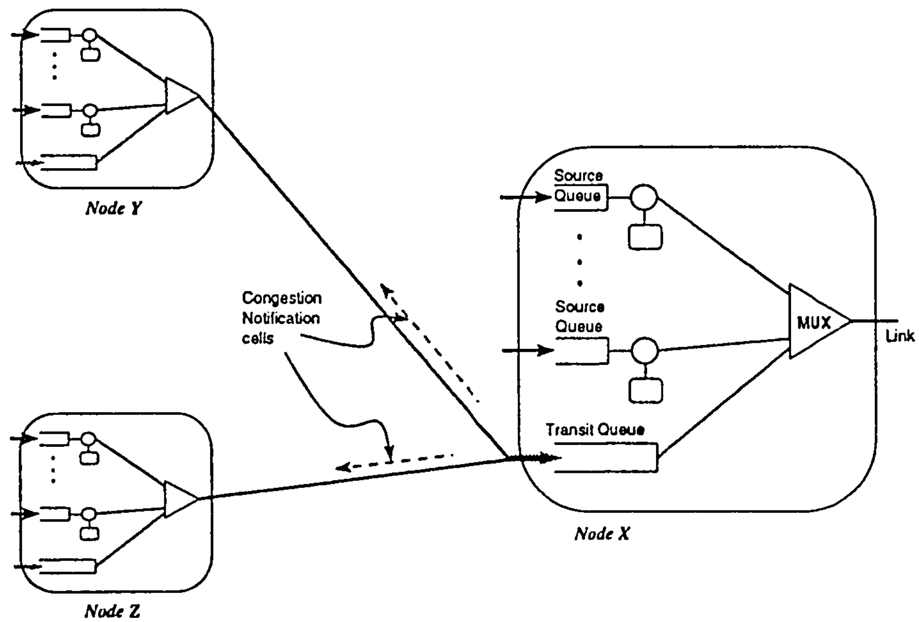

Figure 4.1: Congestion state.

notifications trigger an increase in the token rates. Again, this increase is only for those traffic sources which are transmitting towards the uncongested node.

4.2.1 Leaky Bucket Control at the Sources

The Leaky Bucket traffic control scheme is applied at each source queue. The implementation consists of a token counter that is updated at a variable rate which depends upon the state of the node (congested or otherwise). However, there is a lower bound for this token generation rate, which is equal to the average cell generation rate of the corresponding traffic source. A cell at the head of a source queue is transmitted only if the token counter is greater than zero. Transmission

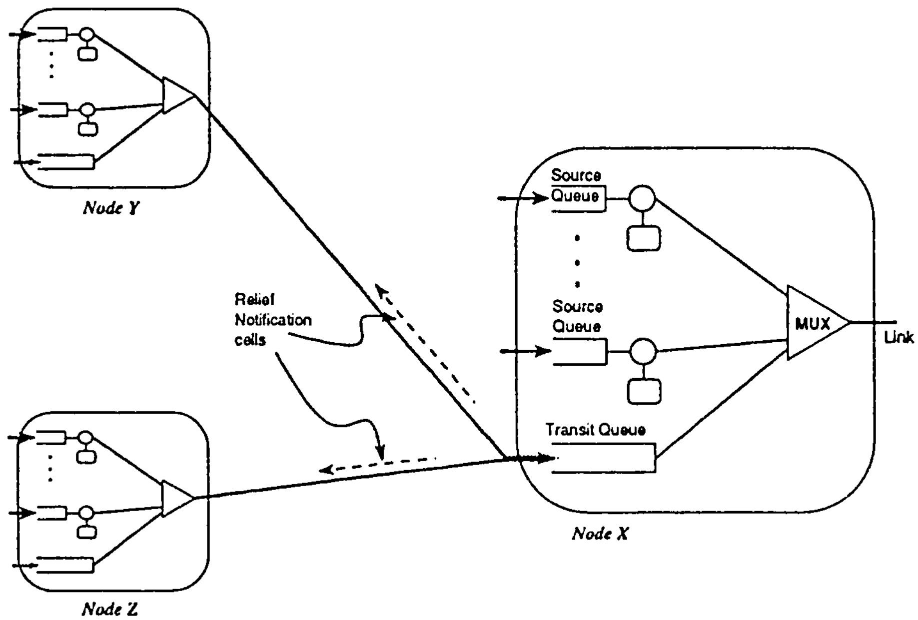

Figure 4.2: Congestion relief state.

of each cell decrements the counter by one. The token counter has an upper bound after which the counter stops incrementing until it is decremented below this limit. This upper limit is set equal to the average burst length of the corresponding traffic source.

This mechanism is a complete traffic control scheme for one particular traffic source. It prohibits the source from generating excessive traffic of cells and thus violating the quality of service (QoS) contract. The lower limit on the token generation rate ascertains that the source does not get deprived in the case of congestion at the node. However, when the traffic load at a node is light the token generation rate can be increased to a maximum value which is equal to the peak cell generation rate of the source.

The magnitude of increase or decrease in the token rate is related to the traffic conditions at a particular node and also at the immediate neighboring nodes of that node.

4.2.2 Backpressure Control

The second component of congestion control scheme implemented in the simulator is called *Backpressure Traffic Control*. It can be considered a variation of the *BECN* scheme. The main advantage of this scheme is its simplicity and fast response time. On the network level, propagation delay is a major factor in the total transmission delay of the cells. Requesting non-compliant sources to throttle down requires relatively long delays. And by the time a source throttles down its traffic, too many cells might have been lost. Furthermore, during this time the congestion state might have been over or worsened to a collapsing degree.

The backpressure traffic control scheme is implemented at each node. This scheme is used to control congestion caused by the traffic passing through a node (transit traffic) and destined to some other downstream nodes. Since the intermediate nodes process each transit cell on an individual basis, they do not have sufficient information to identify the misbehaving traffic source(s). That is why any traffic control measure might result in unfairly penalizing compliant sources along with the non-compliant ones.

This scheme requires only 1 bit in the cell header to notify the congestion to the immediate predecessor nodes. It is assumed that a separate control network exists between all neighboring nodes on which the congestion notification can be sent even if there is no reverse data path between the nodes. This is a valid assumption because all communication networks have such control connections for exchanging operation, administration and maintenance (OAM) information.

Congestion Detection

The triggering of the Backpressure control mechanism requires the detection of congestion state at a node. This detection takes place at the *Transit Queue* which is used for queuing the transit traffic at a node. The transit queue has a fixed congestion threshold level. When the number of cells stored in the transit queue crosses that level, the node enters into the congested state.

Backpressure Notification and Traffic Control

Once a node enters into the congested state, the following steps are taken to control the congestion and bring the node to a non-congested state again.

- The token generation rates of the traffic sources attached to the congested node and transmitting on the same outgoing link are reduced.
- The congested node informs its immediate upstream nodes by sending congestion notification notices at regular intervals as long as congestion persists.

Upon receiving a congestion notification from a downstream node, the recipient node reduces the token generation rate of the traffic sources attached to it. The token rates of only those sources whose traffic is transmitted to the congested node are reduced. The reduction observes the lower limit for each source and the reduction is according to the function described in the next section.

These traffic control measures are only taken by the immediate neighbors of the congested node. The congestion information is not propagated to nodes farther away since this would require much more information to identify the congested node. This would not only take more processing but the effectiveness of such information diminishes as the distance from the congested node increases. Since the congestion is a transient state of relatively short duration (because of the statistical nature of traffic), a delayed action by a farther node might only decrease the throughput of the network.

Congestion Recovery

As mentioned above, the congestion state of a node is a transient state. In most of the cases it takes place due to the statistical nature of the traffic sources. Once the congestion state is over, the traffic rate of the sources can be increased to maximize the network throughput. Congestion recovery is the mechanism required to increase the throughput of the network after the congestion period is over. It is based on the following steps:

1. Congestion relief detection.

2. Congestion relief notification.

3. Token generation rate increase.

Congestion relief detection is also based on a threshold level of the transit queue. This second threshold, which is lower than the congestion detection threshold, detects the relief of congestion at the node.

Congestion relief notification is sent to all the upstream nodes in the same manner as the congestion notification. Control cells are sent to the upstream nodes to indicate the uncongested state of the node. But only one such cell is sent to each upstream node since there is no point in continuously sending such cells.

Token generation rate increase for the sources takes place at both the previously congested node as well as the previously notified upstream nodes. The increase in the rate takes place at regular intervals. As mentioned earlier, this increase in the rate can take place up to the upper limit for each source which is equal to the peak rate of the source.

4.3 Important Parameters

There are a number of parameters which have to be set for an effective implementation of the scheme. The values of these parameters depend upon either the traffic source characteristics, i.e. (mean or peak) cell rate, traffic burstiness and mean burst length, or the link characteristics such as transmission bandwidth and propagation delay.

The following are the important parameters effecting the performance of the scheme along with the methods of obtaining their suitable values.

1. **Source Queue Limit**

   The maximum buffer size of the source queue determines how long a burst can be accommodated by the node from the corresponding source. This queue limit should be large enough to provide the QoS guaranteed to the source by the network. That is, the cell loss from the queue due to overflow should be less than the cell loss tolerance of the source.

   The capacity of a source queue is estimated while making the approximation of modeling a source queue and its associated link as an $M/M/1/K$ queue. Recall that for an $M/M/1/K$ queue, the probability of loss is equal to

$$P_{loss} = \frac{1-\rho}{1-\rho^{K+1}}\rho^{K} \tag{4.1}$$

   where,

   ρ is the utilization of the resource, and

   K is the size of the queue.

Let,

$$
\begin{aligned}
p &= \text{Cell loss tolerance of the traffic source} \\
\rho &= \text{Utilization of the outgoing link} \\
K &= \text{Estimated queue size} \\
t_{on} &= \text{Average ON period of the source} \\
t_{off} &= \text{Average OFF period of the source}
\end{aligned}
$$

Replacing P_{loss} with p in Equation 1, we get

$$p = \frac{1-\rho}{1-\rho^{K+1}}\rho^{K} \tag{4.2}$$

Solving the above equation for K, we get

$$K = \frac{log(p) - log(1+(p-1)\rho)}{log(\rho)} \tag{4.3}$$

For $p << 1$, as in ATM networks, the second term in the numerator becomes negligible. Therefore, we are left with

$$K = \frac{log(p)}{log(\rho)} \tag{4.4}$$

Since the traffic source is ON-OFF source, we multiply Equation 4.4 by the proportion of the time the source is ON to get the source queue capacity. That is,

$$\frac{t_{on}}{t_{on} + t_{off}}$$

But,

$$\frac{t_{on}}{t_{on} + t_{off}} = \frac{1}{\beta} \tag{4.5}$$

where, β is the burstiness of the traffic source. Therefore, multiplying Equation 4.4 with $1/\beta$, we get

$$\begin{aligned} Source\ \ Queue\ \ Capacity &= \frac{1}{\beta} \times K \\ &= \frac{1}{\beta} \times \frac{log(p)}{log(\rho)} \end{aligned} \tag{4.6}$$

The link utilization is assumed to be equal to 75% for estimating the source queue capacities.

2. **Transit Queue Capacity**

Transit queue is used to hold the transiting traffic through a node. Since a node does not have any information about the cells in the traffic, e.g. the location of the generating sources and their characteristics, the buffer size for storing these cells should be enough to minimize the cell loss under all conditions, i.e. congested or uncongested.

The capacity of a transit queue is calculated according to the function given below.

$$Transit\ \ Queue\ \ Capacity = \sum_{i=1}^{k} N_i \tag{4.7}$$

where,

N_i is the average burst length of a VC_i,

k is the total number of VCI's passing through the transit queue.

3. **Congestion Threshold**

The threshold for triggering the congestion control mechanism is set so as not to violate the minimum delay jitter tolerance of any of the connections forwarding traffic via the transit queue of this node. This threshold is estimated as follows.

(i) Let,

$$\begin{aligned}
D &= \text{Network diameter expressed in number of hops minus one} \\
k &= \text{Number of active queues (source queue and the transit queue) using the link} \\
C &= \text{Link capacity expressed in Mbps} \\
\mu^{-1} &= \text{Average service time of an ATM node for transit queue} \\
&= \frac{424}{C/k} \\
J_{min} &= \text{Minimum delay jitter tolerance of all sources connected to the transit queue}
\end{aligned}$$

When the transit queue has M cells, the time spent by an arriving cell in the queue would be,

$$T_h = \frac{M}{\mu} sec. \tag{4.8}$$

T_h is the average cell delay per hop. Hence, the maximum end-to-end delay jitter would be,

$$T_{max} = D \times T_h \tag{4.9}$$

This maximum end-to-end delay must be less than the delay jitter requirements, i.e.,

$$T_{max} \leq J_{min}$$

Therefore, using eq. 4.9

$$\begin{aligned} D \times T_h &\leq J_{min} \\ T_h &\leq \frac{J_{min}}{D} \\ M \times \frac{424}{C/k} &\leq \frac{J_{min}}{D} \\ M &\leq \frac{J_{min}}{D} \times \frac{C}{424 \times k} \end{aligned}$$

That is,

$$M \leq \frac{\alpha}{k} \tag{4.10}$$

where,

$$\alpha = \frac{J_{min} \times C}{D \times 424} \tag{4.11}$$

(ii) Let,

l = Length of the outgoing link

$$
\begin{aligned}
v &= \text{Signal propagation speed of the link} \\
t_{cell} &= \text{Time taken to transmit one cell} = 424/C \\
t_{prop} &= \text{Propagation delay of the link} = l/v \\
a &= \text{Maximum number of cell arrivals during } t_{prop} \\
&= \frac{t_{prop}}{t_{cell}}
\end{aligned}
$$

The number of cell departures from the transit queue during t_{prop} is,

$$d = \frac{a}{k}$$

Hence, the backlog of cells during t_{prop} is

$$
\begin{aligned}
\text{Backlog } (t_{prop}) &= a - \frac{a}{k} \\
&\approx a \text{ for large } k
\end{aligned}
\tag{4.12}
$$

The reduced rates from upstream nodes will be felt at the congested node $2t_{prop}$ from the time the node sends the congestion notification. Therefore, we set the congestion threshold as follows,

$$\text{Congestion Threshold} = M - 2a \tag{4.13}$$

Equation 4.13 is used to estimate the transit queue threshold at which the congestion alarm should be triggered for a particular node.

4. **Congestion Relief Threshold**

   The threshold for triggering the congestion relief mechanism is set equal to zero i.e, the congestion relief measures will be triggered only when the transit queue becomes empty after entering the congestion state.

5. **Token Rate Reduction Factor**

   Upon entering into the congestion state or on arrival of a congestion notification, the token rates of the source queues are reduced. Multiplicative decrease is used to cut the token rate of the source queues. It has been proved in [10] that multiplicative decrease and additive increase is the optimal strategy for traffic control.

   The reduction factor for the source queues is one half of the previous value of the Token Rate, i.e.

$$New\_Rate = min(m, \frac{Current\_Rate}{2}) \tag{4.14}$$

   The choice of the exponential reduction factor is based on the reduction factor used in the TCP/IP protocols to control the traffic flow.

6. **Token Rate Increase Factor**

   For a previously congested node, once congestion is thought to be over, the traffic sources attached to the node are allowed to increase their rates by increasing their token generation rates. Also, those traffic sources attached to the immediate predecessor nodes which forward traffic via the node in question are allowed to increase their token generation rates. The token generation rates are increased according to the following function.

$$New\_Rate = Present\_Rate + \frac{\beta - 1}{\beta} \times m \tag{4.15}$$

where,

β is the burstiness of the source, and

m is the mean data rate of the source in bits per second.

The choice of the function is made so that the additive factor is more for sources with high burstiness factor as compared to sources with lower burstiness. Figure 4.3 shows the additive factor for different values of β. Therefore, for highly bursty sources, the rate is augmented in increments of m, whereas for less bursty sources the rate is augmented in a small fraction of m.

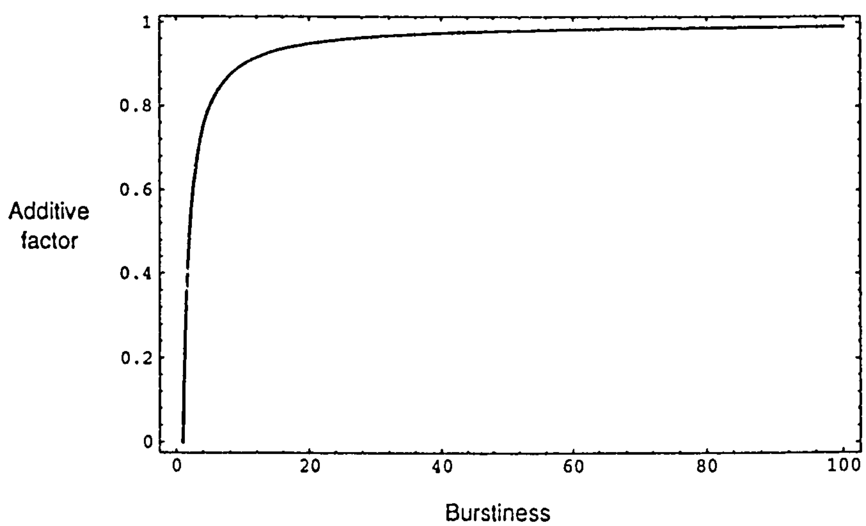

Figure 4.3: Additive factor vs. Burstiness.

Chapter 5

Simulation Model

From a traffic engineering perspective a computer communication network is basically a queuing network. Such a network is a collection of nodes (service providing entities) connected in such a way that cells (customers, in general queuing terminology) proceed from one node to the next on their way from a source to a destination.

There are two approaches that are generally used to analyze such systems;

- Markovian Queuing Theory approach and its variants
- Simulation

The assumptions necessary for the application of the first approach to ATM networks in general, and to the problem of traffic control in particular, makes it impractical for obtaining applicable solutions. Therefore, we have chosen simulation for analysis of the traffic control scheme proposed in this thesis. Since activities happen at discrete points in time in data communication network, *Discrete-event* simulation is a common approach to simulate data communication networks. This approach is also used in this thesis to simulate an arbitrary ATM network and to analyze the scheme.

Different simulation tools are available to model communication networks in a graphical manner. But the large amount of data and the number of events generated in an ATM network, especially when modelled at the cell level, makes their use restrictive to very small networks. Therefore, in this thesis, we implemented a software program capable of simulating any user specified ATM network where our traffic congestion scheme is adopted. The program is implemented in the C language. This chapter describes the simulation model and the assumptions made about the network. The output results of the simulations are discussed in the next chapter.

5.1 Assumptions about the Network

The simulator is designed with the following assumptions about the network.

- The Virtual Connections are already established and have passed the *Admission Control* phase.
- The virtual connections remain established throughout the duration of the network simulation.
- The transmission from a node takes place in a *Round Robin* manner.

- There are no loops in the virtual connection, i.e. a node is not visited more than once along the path of the virtual connection.

5.2 Elements of the Simulation Model

The node model in the simulator contains the data structure which links the different components in the node together. The main components of a node are:

- the sources,
- the communication links, and
- the queues.

5.2.1 The Source Model

The traffic source model assumed is the ON-OFF model. This is the most widely used source model in ATM studies. The details of this model were given in chapter 2. The ON-OFF model requires the generation of exponentially distributed random numbers. For this purpose a random number generator with uniform distribution is required. This simulation uses the uniform random number generator given in [60]. This random number generator has 100 starting streams. The uniformly distributed random numbers are converted into exponentially distributed number using the following inverse-transform algorithm:

1. Generate $U \sim$ U(0,1).

2. Return $X = -\beta \ln U$.

where, U is a uniformly distributed random number between 0 and 1,

x is an exponentially distributed number with average $1/\beta$.

and $\beta > 0$ is the rate of the exponential distribution for which the random number is required.

5.2.2 The Link Model

The links are the main cause of delay in ATM networks. The transmission links in most of the MAN/WAN ATM networks are optical fibers. The signal speed on these fibers is equal to about 80% of the speed of light i.e. $2.3 \times 10^8 m/s$. Usually the bit error rate on such fibers is of the order of 10^{-12}. In the simulator developed, the links are inherently modelled to be optical fibers. Due to the negligible bit error rate of optical fibers and considering the single error correction capability of the ATM cell header checksum, the links are modelled as error free transmission channels.

The transmission capacity of the links, though, can vary depending upon the material characteristics of the fiber. In present day optical fibers, transmission capacity of 155 Mbps (OC-3) on long haul trunks is common. 622 Mbps (OC-12) capacity fibers are now being used on longer distances and heavily loaded links. Higher capacity fibers, in Gbps range, are currently in the experimental phase. Due to the availability of different options in link capacity, this characteristic of the links has been kept as an input parameter of the simulator.

5.2.3 The Queue Models

The queues are used at the nodes to store the ATM cells that can not be transmitted immediately because of the link occupancy by other cells. Queues can be of different types but the most commonly used type of queue in ATM networks is the first-in-first-out (FIFO) queue. The reason for the wide use of this type of queue is its simplicity. Other types of queues require more processing time at the node which can contribute to congestion.

Two kinds of FIFO queues have been modelled for each node. These are described below.

Source Queue Model

A source queue is used for queuing the cells generated by a source attached to a node. One source queue is used for each attached source (Figure 5.1). The size of the queue depends upon the traffic source parameters and is determined using Equation 4.6.

The cells queued at a source queue are served in a FIFO manner, where the service rate is dictated by the Leaky Bucket congestion control algorithm (Leaky Bucket control implementation is discussed later in the chapter). The simulator

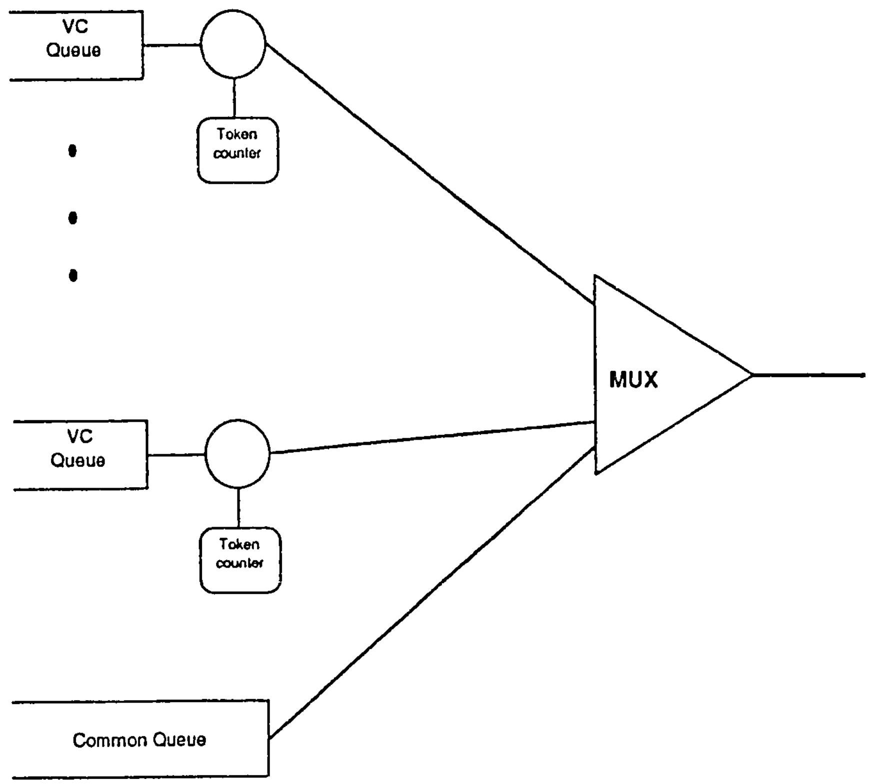

Figure 5.1: Node structure for one link.

keeps track of the average size of each such queue in terms of the number of cells. It also counts the total number of cells dropped at each queue because of overflow.

Transit Queue Model

Each node has another FIFO queue, called the *Transit queue.* This queue is used to accommodate excess cell traffic transiting through the node from some upstream source nodes to some downstream destination nodes (Figure 5.1).

The main difference between the Source Queue model and the Transit Queue model is the congestion control method used at each kind of node. As mentioned earlier, source queue model employs leaky bucket algorithm to control congestion at the queue. In a transit queue no congestion control method is applied directly to the queue. Instead, thresholds are used to detect congestion at a particular transmit queue. This detection is then used to implement the *backpressure* congestion control technique described earlier. The main reason for not applying the leaky bucket algorithm to transit queues is that a transit node has no way of efficiently identifying the traffic parameters of all the sources which are generating the transit traffic and adjusting the parameters of the leaky bucket appropriately. If such a source has to be identified then each node has to process all transiting traffic which would considerably increase the processing load of the node. This situation in itself can become a major cause of congestion. Since statistical multiplexing is used in ATM networks, there might be a scenario that congestion at a particular node is the effect of many sources transmitting simultaneously near or at their peak rate without any excessive transmissions from one source. Therefore, in general, several sources can be responsible for the congestion. Hence, identifying only one source and requesting it to reduce will not be sufficient in most cases. Identifying all such sources might increase the processing overhead to a level where this would in itself become cause of congestion.

One transit queue is modelled for each outgoing link at each node. Two threshold occupancy levels are used at each node, one to detect congestion and thus to trigger

Input parameter	Reserved Word
Node	.Node
Link	.Link
Source	.Source
Virtual Connection	.VC
Simulation Time	.Time
Comment	#

Table 5.1: Input parameters and their reserved words.

the control mechanism, and the other to detect the end of congestion period. As in the case of source queue, the average queue length and the number of cells dropped at each transit queue are also calculated.

5.3 Input Parameters

The simulator is written in a manner so that different networks can be simulated without changing the data structure or code for every network. For this purpose, the essential parameters that describe an ATM network are read by the simulator from an input file. The input file is a text (ASCII) file with reserved words used to indicate the various elements and parameters of the network. The input data structures are then filled accordingly.

The simulator requires five input parameters to be given in the input file. These are given along with their reserved words in Table 5.1.

5.3.1 Node parameters

A node in a communication network is a collection of data sources and/or sinks. The sinks need not be specified since they are mentioned as the last node of a virtual connection (Virtual Connection parameters are described below). The basic parameters in the description of a node are the number of sources attached to that node.

The reserved word for beginning the description of nodes is *".Node"*. The description starts from the next new line and has the following sequence of parameters.

```
label no_of_sources
```

where,

`label` is the label of the node, and

`no_of_sources` is the total number of data sources attached to the node.

5.3.2 Source parameters

Sources have been modeled as ON-OFF sources (details of source modeling are explained in the next section). The parameters required to describe such a source as recommended by the International Telecommunication Union (ITU) are as follows:

- The average or the peak cell generation rate.
- The mean burst length.

- The burstiness factor (which is the ratio of the peak to mean cell rate).

The input file contains these three parameters for each source. To identify each source, a separate label is given to each one. In addition to these parameters, the simulator also requires the knowledge of the node to which a particular source is attached. Therefore, each source description also contains the label of the node to which it is connected.

All sources of the network are described after the reserved word *".Source"* starting from the next line. The sequence of parameters is as follows.

```
label node_label avg_rate mean_burst_length burstiness
```

where,

`label` is the label of the source,

`node_label` is the label of the node the source is connected to,

`avg_rate` is the mean cell generation rate of the source (in cells/sec),

`mean_burst_length` is the average length (in cells) of a burst for the source, and

`burstiness` is the burstiness factor for the source, i.e the ratio of the peak to the mean cell rate.

5.3.3 Link parameters

ATM networks usually have fiber optic links connecting different nodes. But for the sake of generality, link capacity is kept as an input parameter instead of a coded constant. The other main parameters describing a link are the following:

- The length of the link.
- The labels of the nodes connected by the link.

It should be noted that the simulator is designed for ATM networks in the MAN/WAN category of networks where nodes are connected by point-to-point links of considerable distances which accounts for long transmission delays. Such a situation is not common in networks over short geographical distances such as LANs.

Links are described in the input file after the reserved word *".Link"* starting from the next line. The sequence of parameters is as follows.

```
label node_label_1 node_label_2 capacity length
```

where,

`label` is the label of the link,

`node_label_1` is the label of the node at the first connection point,

`node_label_2` is the label of the node at the other connection point,

`capacity` is the bit rate of the link (in Mbps), and

`length` is the length of the link between the two connected nodes (in kilometers).

5.3.4 Virtual Connection parameters

Virtual connections (VCs) are the basic connections between two points in ATM networks. The simulator developed in this thesis simulates the network in a mode where the connections are already established i.e. it does not initiate new connections during a simulation run. Therefore, these connections are to be provided to the simulator as an input. Since the path of a connection is determined at the connection setup time, which can depend on the admission control algorithms used by the network, these connections have been left as an input option to the simulator. This also provides the possibility of analyzing the effect of different paths for a particular connection on the network.

Virtual connections are identified in the input file by the reserved word *".VC"*. The description of different VCs are given in the next line after the *".VC"* keyword. The sequence of parameters is as follows.

```
label src_label nodes_spanned org_node [int_nodes] dest_node
```

where,

`label` is the label (or VCI) of the VC,

`src_label` is the label of the data source for the VC,

`nodes_spanned` is the total number of nodes, including the source and the sink nodes, through which the cell of the VC passes,

`org_node` is the label of the originating node for the VC,

`int_nodes` are the labels of the intermediate nodes that the VC passes through to connect to the destination node, and

`dest_node` is the label of the destination node for the VC.

5.3.5 Logical Simulation Time parameter

This parameter indicates for how long the network is to be simulated. In this manner a network can be simulated for different time spans. This parameter is described in the line after the reserved word *".Time"*. The time can be given from hours up to micro-seconds. The input sequence is as follows.

```
hours minutes seconds milliseconds microseconds
```

where,

`hours` is the number of hours to be simulated,

`minutes` is the number of minutes to be simulated,

`seconds` is the number of seconds to be simulated,

`milliseconds` is the number of milliseconds to be simulated, and

`microseconds` is the number of microseconds to be simulated,

5.4 Simulator Output

The simulator produces one output data file for each queue in the network. In addition to this, one output data file per node is also generated to show the status of the links at that node. The output data is generated periodically. This period is a user specified parameter. At the end of the simulation, an output file is generated which contains the average values of important parameters for each queue and link.

The output files are in ASCII text format which can be used to produce different plots using any plotting software tool which accepts ASCII format as input.

5.4.1 Source Queue Output File

The output data file for each source queue contains both the parameters of the source and the estimated performance variables for the source queue. The values of the parameters are written at the beginning of the file. This is followed by the observed values of the performance variables.

The following parameters are listed in the output file.

1. Queue capacity
2. Maximum token counter value
3. Mean rate of the source
4. Mean burst length of the source
5. Burstiness factor
6. Cell loss tolerance of the source
7. Cell delay tolerance of the source

The values of the following variables are recorded in the output file after regular time intervals.

1. Simulation clock value
2. Queue size
3. Number of tokens
4. Token rate
5. Number of cells arrived at the queue
6. Number of cells dropped at the queue

5.4.2 Transit Queue Output File

The output file for each transit queue also contains the values of both the constant parameters and the time varying performance metrics of the queue. The format is similar to that of the source queue output file.

The output data file for each transit queue contains the values for the following constant parameters.

1. Queue capacity
2. Congestion notification threshold
3. Relief notification threshold
4. Number of VCI's utilizing the queue

The time varying performance metrics provided in the output file are the following.

1. Simulation clock value
2. Queue size
3. Number of cells arrived at the queue
4. Number of cells dropped at the queue

5.4.3 Links Output File

For each node, one output file is generated which shows two parameters about each link connected to the node along with a time stamp. These parameters are:

1. Congestion status
2. Activity status

The congestion status shows whether the link is congested or not at the instant when the snapshot was recorded. The activity status shows whether the link is idle or busy in the transmission of a cell at that time instant.

5.4.4 Average Values Output File

The average values of the following important parameters are calculated at the end of the simulation.

1. Average delay for all queues

2. Average queue length

3. Link utilization for all links

4. Average service time at each node

This file also contains the total time taken by the simulation to run and the actual (logical) time simulated on the simulator. Total number of cells generated during the simulation is also stated.

5.5 An Example

Figure 5.2 shows a typical ATM network of five nodes with six links. The bandwidth of the links and their lengths are also shown in the figure beside each link.

The input file describing this network is as follows.

```
.Source
# label node_label avg_rate mean_burst_length burstiness
1 1 10 2500 10
2 1 20 9660 1000
3 1 .016 47 10
4 2 10 950 1000
5 2 40 2400 5
3 .005 28 100
7 3 .064 154 1
8 4 1 1900 5
9 5 5 5 40
10 5 .096 38 100
```

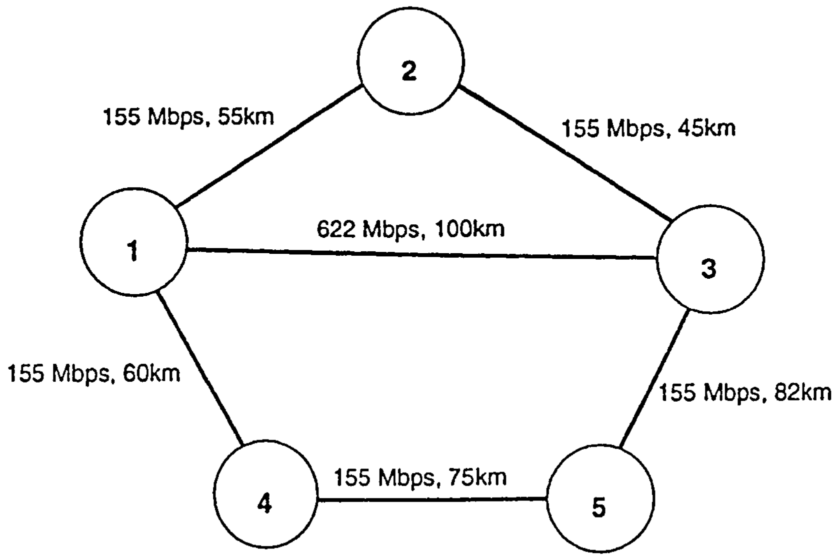

Figure 5.2: An example network.

```
.Node
# label no_of_sources
1 3
2 2
3 2
4 1
5 2

.Link
# label node_1 node_2 capacity length
1 1 2 155 50
2 2 3 155 75
3 1 3 622 100
4 1 4 155 45
5 4 5 155 35
6 3 4 155 60

.VC
# label src_label nodes_spanned org_node [int_nodes] dest_node
1 1 3 1 3 4
```

```
2 5 3 2 1 5
3 2 3 1 2 3
4 3 4 1 5 4 3
5 4 2 2 3
6 7 4 3 4 5 1
7 8 2 4 3
8 9 2 5 4
9 10 3 5 1 2
10 6 2 3 4

.Time
# hours min sec msec microsec
0 0 10 10 19
```

Chapter 6

Simulation Results and Discussion

This chapter presents the performance of the proposed scheme in the light of the results obtained from the developed simulator for two network topologies. The results have been compared with the simple *Leaky Bucket* congestion control technique to show the effectiveness of the reactive token rate control. The average results obtained at the end of simulation were used to generate several plots.

The performance of the scheme is discussed at the end of the chapter in light of the four key criteria [10]:

1. Efficiency

2. Fairness

3. Distributedness

4. Convergence

The definitions of these criteria have already been given in Chapter 1. Because of excessive simulation run times, we have limited our experiments to two networks only. We refer to these networks as Network_1 and Network_2 respectively. Below we summarize the experimental data collected from our simulation runs.

6.1 Network_1

Figure 6.1 shows the topology and connections of the first network on which the scheme has been tested. The network consists of five nodes and six links. Table 6.1 lists the capacity and length of the each link. Tables 6.2 and 6.3 show the parameters of the traffic sources used for different load conditions. These parameters have been chosen from [21]. Table 6.4 shows the delay jitter tolerances and cell loss tolerances for these traffic sources. To observe the performance of the scheme under different load conditions, the average loads on the links were varied from 50 to 140 Mbps.

Table 6.1 shows the average delays observed at the source queues for different traffic loads. Table 6.1 shows the average delays observed at the transit queues for different load conditions. It can be observed from the table that the average delay increases with the increase in the traffic load. But the delay does not exceed the minimum delay jitter tolerance of any source which is equal to 10 msec.

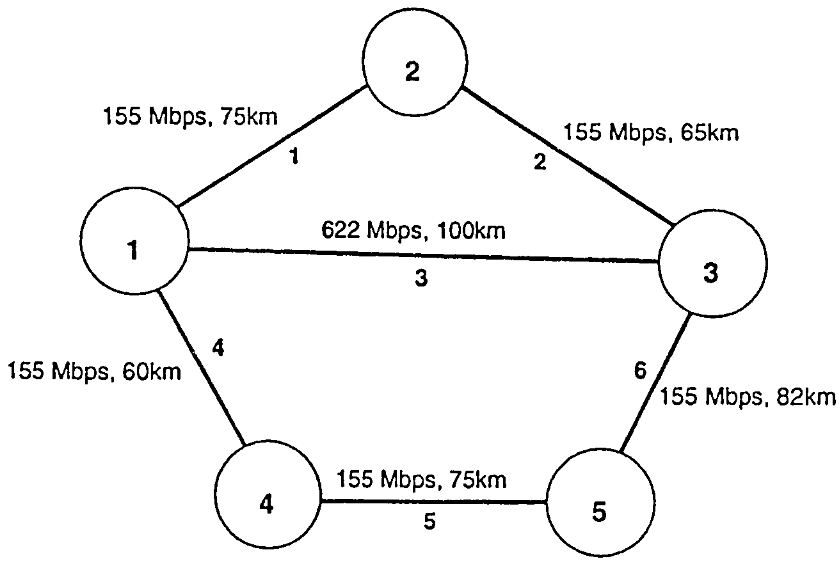

Figure 6.1: Network_1.

Link No.	Capacity (in Mbps)	Length (in km.)
1	155	75
2	155	65
3	155	100
4	155	60
5	155	75
6	155	80

Table 6.1: Link capacities and lengths for Network_1.

Source No.	Avg. Burst Length	Load 1 Av. Rate (in Mbps)	Load 1 β	Load 2 Av. Rate (in Mbps)	Load 2 β	Load 3 Av. Rate (in Mbps)	Load 3 β	Load 4 Av. Rate (in Mbps)	Load 4 β
1	1000	5	2	7	3	10	4	13	5
2	50	2	5	3	7	4	9	5	11
3	500	10	2	12	5	15	7	20	8
4	300	6	6	10	10	13	12	17	13
5	1200	15	3	20	5	20	7	24	7
6	200	5	8	8	10	10	10	15	15
7	800	10	5	13	8	15	9	18	10
8	1300	20	4	25	7	30	9	35	10
9	100	1	10	5	12	7	15	9	15
10	1500	10	3	11	6	14	7	17	8

Table 6.2: Traffic sources for different load conditions on Network_1.

Source	Load 5 Av. Rate (in Mbps)	Load 5 β	Load 6 Av. Rate (in Mbps)	Load 6 β	Load 7 Av. Rate (in Mbps)	Load 7 β	Load 8 Av. Rate (in Mbps)	Load 8 β
1	15	6	19	7	23	8	27	9
2	6	11	7	12	8	12	9	12
3	24	10	25	11	27	12	32	12
4	20	14	23	14	26	15	31	15
5	29	8	32	8	35	9	37	9
6	18	13	20	10	24	7	28	5
7	23	11	26	12	30	13	34	14
8	40	12	45	15	50	15	55	15
9	13	16	15	17	17	17	19	18
10	19	9	21	10	25	11	27	12

Table 6.3: Traffic sources for different load conditions on Network_1.

Source No.	Delay Jitter Tolerance (in msec.)	Cell Loss Tolerance
1	70	10^{-10}
2	10	10^{-6}
3	10	10^{-10}
4	100	10^{-9}
5	10	10^{-10}
6	50	10^{-9}
7	80	10^{-9}
8	20	10^{-10}
9	10	10^{-7}
10	10	10^{-10}

Table 6.4: Delay jitter and cell loss tolerances for Network_1.

Source	Average delay (in msec.)							
Queue	Load 1	Load 2	Load 3	Load 4	Load 5	Load 6	Load 7	Load 8
1	3.25	3.73	4.18	4.81	5.54	6.14	6.92	7.63
2	3.31	3.75	4.22	4.85	5.57	6.2	6.97	7.71
3	3.27	3.72	4.21	4.82	5.55	6.19	6.93	7.65
4	2.81	3.25	3.82	4.43	5.1	5.63	6.32	6.91
5	2.77	3.11	3.75	4.39	5.01	5.58	6.19	6.82
6	3.12	3.73	4.2	4.66	5.11	5.59	6.2	6.81
7	3.08	3.62	4.08	4.61	5.09	5.5	6.17	6.79
8	1.98	2.51	3.03	3.61	4.08	4.54	4.92	5.33
9	2.85	3.31	3.9	4.47	4.94	5.42	6.01	6.53
10	2.97	3.45	4.11	4.56	5.07	5.38	5.89	6.47

Table 6.5: Average delays of source queues for Network_1.

Transit Queue	Average delay (in msec.)							
	Load 1	Load 2	Load 3	Load 4	Load 5	Load 6	Load 7	Load 8
1	2.98	3.51	3.94	4.67	5.09	5.77	6.23	6.72
2	3.02	3.55	4.01	4.71	5.13	5.83	6.28	6.76
3	2.91	3.49	3.98	4.65	5.15	5.86	6.31	5.83
4	2.54	2.91	3.61	4.19	4.78	5.44	5.87	6.41
5	2.61	2.96	3.54	4.12	4.81	5.47	5.95	6.51
6	2.81	3.48	3.91	4.48	5.03	5.65	6.1	6.63
7	2.79	3.51	3.93	4.44	4.97	5.51	5.91	6.5
8	2.77	3.52	3.98	4.51	5.11	5.58	6.02	6.48
9	1.62	2.05	2.56	3.13	3.86	4.42	4.83	5.25
10	1.65	2.11	2.61	3.17	3.79	4.39	4.8	5.24
11	2.41	3.13	3.7	4.32	4.84	5.25	5.76	6.29
12	2.52	3.19	3.75	4.28	4.81	5.24	5.81	6.34

Table 6.6: Average delays of transit queues for Network_1.

The values of the following important variables were also observed for the different load conditions. These variables are plotted against the various load values in Figures 6.2 to 6.5.

- Average link utilization
- Average delay jitter
- Percentage cell loss
- Percentage congested time

Load	Link Utilization						Av. util
	Link 1	Link 2	Link 3	Link 4	Link 5	Link 6	
1	0.52	0.48	0.55	0.49	0.47	0.49	0.5
2	0.55	0.52	0.58	0.49	0.49	0.52	0.525
3	0.62	0.57	0.65	0.54	0.56	0.59	0.588
4	0.68	0.63	0.7	0.61	0.62	0.66	0.65
5	0.72	0.69	0.75	0.68	0.67	0.71	0.7
6	0.76	0.75	0.79	0.74	0.76	0.77	0.761
7	0.82	0.8	0.83	0.79	0.81	0.81	0.81
8	0.85	0.84	0.86	0.82	0.83	0.84	0.84

Table 6.7: Link utilizations for Network_1.

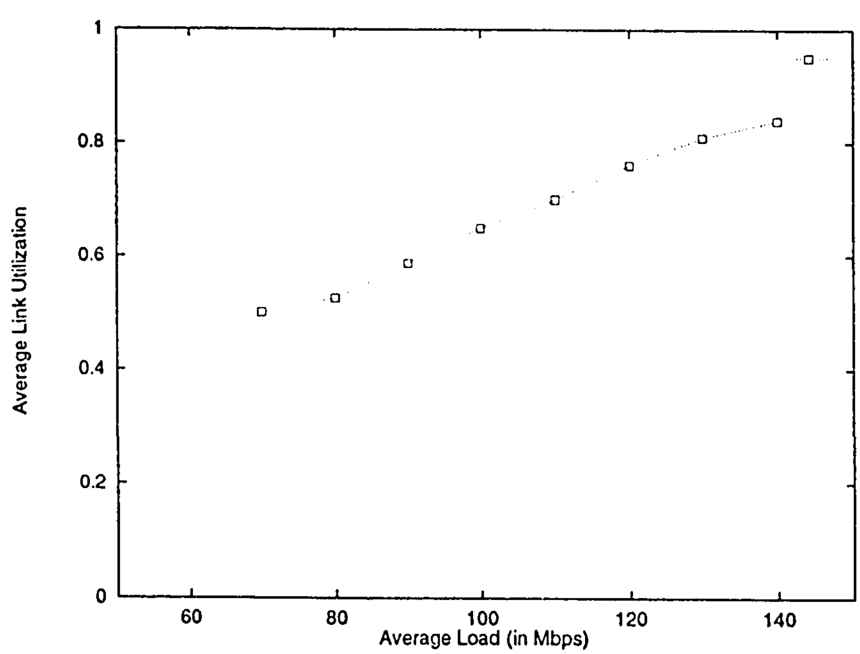

Figure 6.2: Average link utilization vs. Average load.

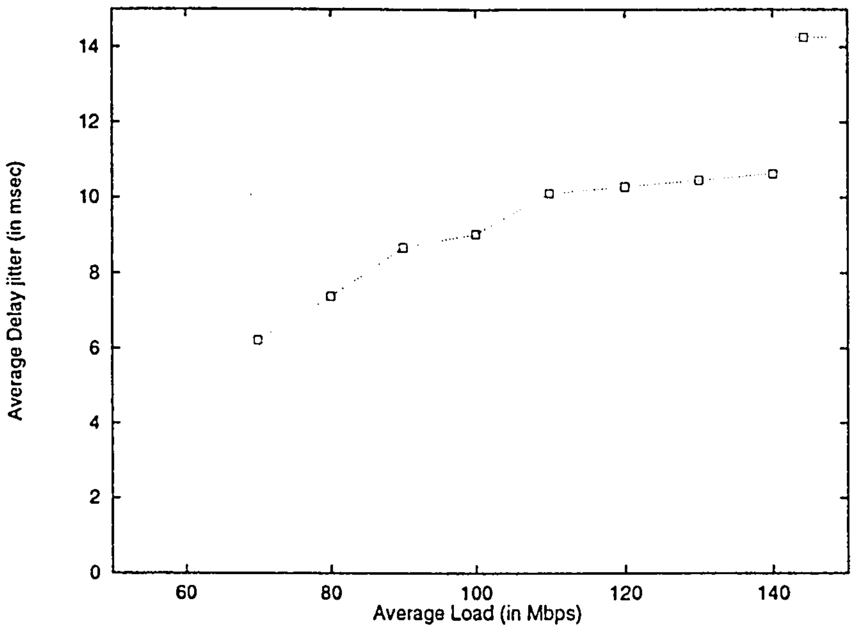

Figure 6.3: Average delay jitter vs. Average load.

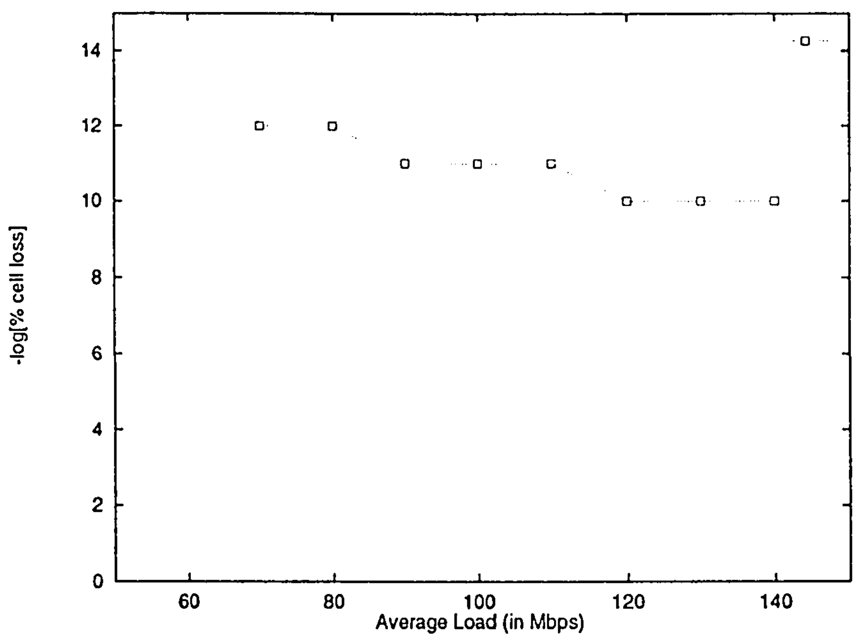

Figure 6.4: Percentage cell loss vs. Average load.

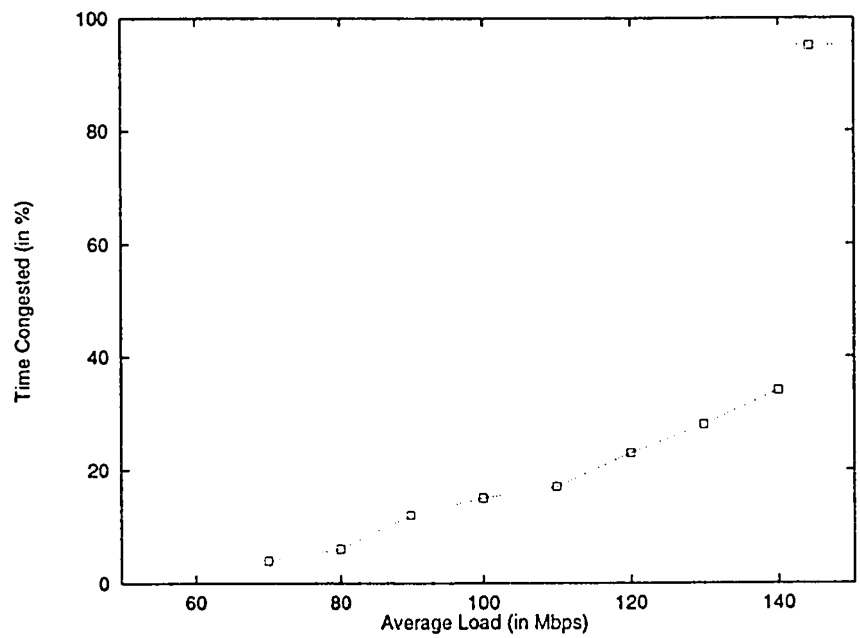

Figure 6.5: Percentage congested time vs. Average load.

6.2 Network_2

Figure 6.2 shows the topology and connections of the second network on which the scheme has been tested. This network consists of four nodes and as many links connecting them in a ring topology. Table 6.8 shows the capacity and lengths of the different links. Tables 6.9 and 6.10 show the traffic sources used for different load conditions. Table 6.11 shows the delay jitter and cell loss tolerances for these traffic sources.

Table 6.2 shows the average delays observed at the source queues for different traffic loads. Table 6.2 shows the average delays observed at the transit queues for different load conditions. It can again be observed from the table that the average

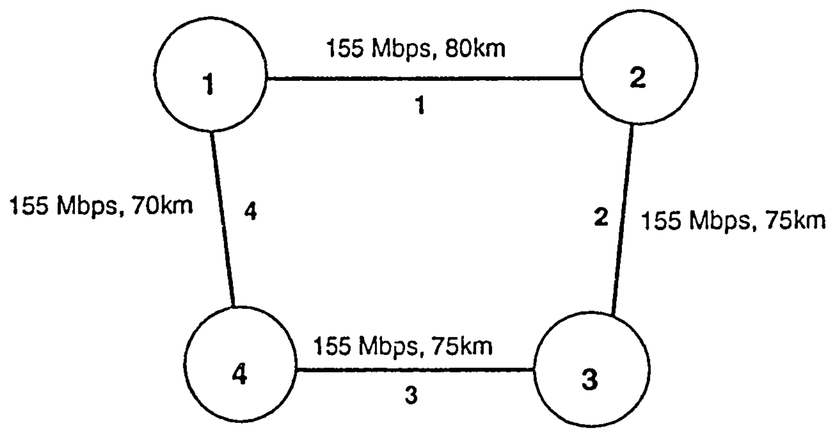

Figure 6.6: Network_2.

Link No.	Capacity (in Mbps)	Length (in km.)
1	155	80
2	155	75
3	155	75
4	155	70

Table 6.8: Link capacities and lengths for Network_2.

Source No.	Avg. Burst Length	Load 1 Av. Rate (in Mbps)	Load 1 β	Load 2 Av. Rate (in Mbps)	Load 2 β	Load 3 Av. Rate (in Mbps)	Load 3 β	Load 4 Av. Rate (in Mbps)	Load 4 β
1	50	4	8	5	7	6	7	7	6
2	500	10	4	12	5	14	6	16	7
3	150	7	10	8	10	9	10	10	10
4	1000	10	3	13	4	15	5	17	6
5	1500	10	2	12	3	13	4	14	5
6	100	2	10	4	12	6	14	8	12
7	250	5	5	7	6	9	7	11	8
8	500	8	5	10	7	12	9	14	11
9	1500	7	4	9	6	11	8	13	10
10	500	7	4	10	5	13	6	16	7
11	600	5	8	7	10	9	8	11	7
12	100	3	10	6	8	9	7	12	6
13	200	4	8	8	10	12	12	16	10
14	750	6	10	7	9	8	8	9	7
15	2000	10	5	15	6	20	7	25	8

Table 6.9: Traffic sources for different load conditions on Network_2.

delay increases with the increase in the traffic load. The delay for this network is higher as compared to Network_1 because the traffic load on this network is relatively heavier. But the delay does not exceed the minimum delay jitter tolerance of any source which is equal to 10 msec.

Figures 6.7 to 6.10 show the plots of average link utilization, average delay jitter, percentage cell loss and percentage congested time respectively.

Source	Load 5		Load 6		Load 7		Load 8	
	Av. Rate (in Mbps)	β	Av. Rate (in Mbps)	β	Av. Rate (in Mbps)	β	Av. Rate (in Mbps)	β
1	8	5	9	4	10	4	10	4
2	18	6	20	5	22	4	24	3
3	11	10	12	10	13	10	14	10
4	19	7	21	6	23	5	25	4
5	15	6	16	7	17	8	18	9
6	10	10	11	9	12	8	13	7
7	13	9	15	8	17	7	19	6
8	16	9	18	7	20	5	23	3
9	15	8	17	6	19	5	20	4
10	19	8	22	9	24	8	25	7
11	13	7	15	7	17	6	19	6
12	15	7	18	6	21	5	24	5
13	20	9	24	8	28	7	32	6
14	10	6	11	5	12	4	13	3
15	30	9	35	10	40	9	45	8

Table 6.10: Traffic sources for different load conditions on Network_2.

Source No.	Delay Jitter Tolerance (in msec.)	Cell Loss Tolerance
1	20	10^{-7}
2	10	10^{-9}
3	90	10^{-9}
4	10	10^{-10}
5	40	10^{-10}
6	10	10^{-6}
7	15	10^{-9}
8	20	10^{-9}
9	100	10^{-8}
10	30	10^{-9}
11	40	10^{-10}
12	50	10^{-9}
13	10	10^{-9}
14	10	10^{-10}
15	70	10^{-10}

Table 6.11: Delay jitter and cell loss tolerances for Network_2.

Source Queue	Average delay (in msec.)							
	Load 1	Load 2	Load 3	Load 4	Load 5	Load 6	Load 7	Load 8
1	3.42	3.91	4.48	5.12	5.69	6.31	6.92	7.51
2	3.51	3.96	4.51	5.16	5.72	6.37	6.94	7.55
3	3.41	3.88	4.45	5.09	5.65	6.29	6.9	7.47
4	3.53	3.97	4.53	5.19	5.74	6.38	6.96	7.56
5	3.38	3.81	4.38	4.94	5.51	5.98	6.57	7.24
6	3.42	3.85	4.4	4.99	5.56	6.05	6.68	7.29
7	3.43	3.87	4.41	5.03	5.59	6.12	6.72	7.31
8	3.47	3.92	4.44	5.07	5.63	6.13	6.76	7.38
9	3.37	3.76	4.36	4.91	5.42	5.88	6.43	7.15
10	3.36	3.8	4.4	4.93	5.47	5.91	6.46	7.16
11	3.31	3.74	4.33	4.87	5.37	5.82	6.37	7.03
12	3.23	3.71	4.28	4.83	5.34	5.8	6.33	6.98
13	3.39	3.81	4.31	4.85	5.31	5.81	6.38	7.09
14	3.43	3.9	4.39	4.9	5.35	5.82	6.44	7.11
15	3.49	3.92	4.4	4.92	5.37	5.89	6.47	7.13

Table 6.12: Average delays of source queues for Network_2.

Transit Queue	Average delay (in msec.)							
	Load 1	Load 2	Load 3	Load 4	Load 5	Load 6	Load 7	Load 8
1	3.4	3.85	4.38	4.96	5.59	6.25	6.84	7.41
2	3.39	3.81	4.37	5.02	5.64	6.27	6.88	7.44
3	3.35	3.77	4.33	4.89	5.44	5.92	6.49	7.15
4	3.37	3.8	4.35	4.92	5.47	5.96	6.53	7.2
5	3.25	3.69	4.28	4.78	5.31	5.78	6.28	6.91
6	3.32	3.73	4.32	4.82	5.38	5.81	6.3	6.95
7	3.37	3.81	4.34	4.8	5.29	5.75	6.34	7.03
8	3.42	3.84	4.36	4.82	5.33	5.78	6.37	7.08

Table 6.13: Average delays of transit queues for Network_2.

Load	Link Utilization				Av. util.
	Link 1	Link 2	Link 3	Link 4	
1	0.54	0.51	0.52	0.55	0.53
2	0.57	0.56	0.55	0.58	0.565
3	0.61	0.59	0.61	0.62	0.61
4	0.67	0.65	0.68	0.69	0.672
5	0.71	0.7	0.73	0.73	0.718
6	0.79	0.75	0.79	0.8	0.783
7	0.82	0.8	0.81	0.82	0.813
8	0.86	0.84	0.85	0.86	0.852

Table 6.14: Link utilizations for Network_2.

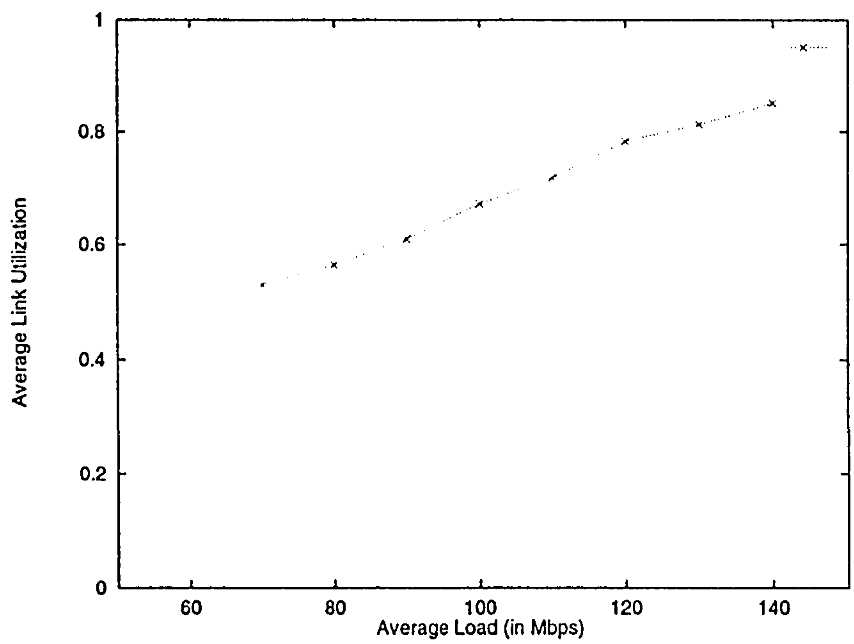

Figure 6.7: Average link utilization vs. Average load.

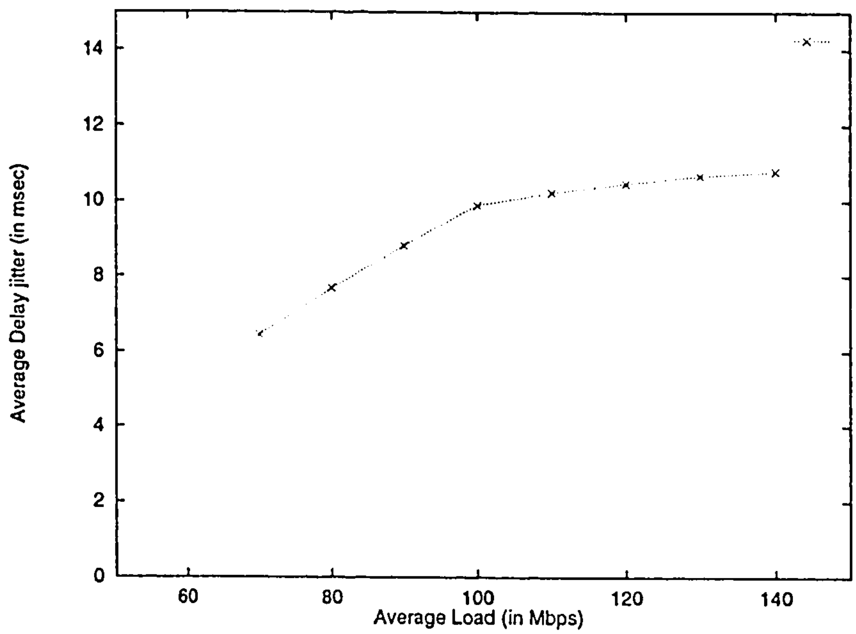

Figure 6.8: Average delay jitter vs. Average load.

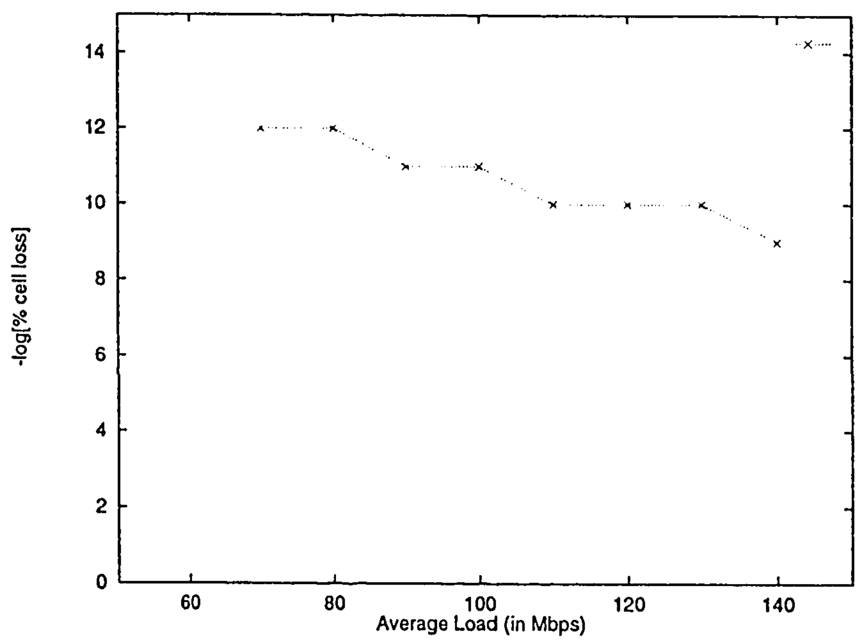

Figure 6.9: Percentage cell loss vs. Average load.

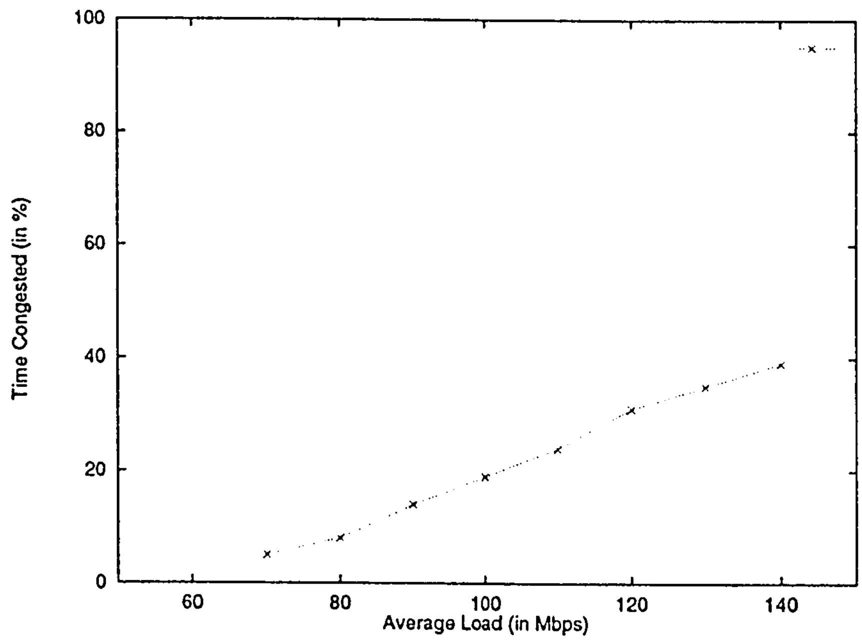

Figure 6.10: Percentage congested time vs. Average load.

6.3 Comparison with Simple Leaky Bucket Scheme

The performance of the scheme is compared with the performance of the *simple* Leaky Bucket (LB) scheme. Two versions of simple leaky bucket are compared for performance with the proposed *Queue Length Based Reactive* ($QLBR$) scheme.

1. Leaky bucket with token rate equal to the peak cell rate of the traffic source (LB_{peak}).

2. Leaky bucket with token rate equal to the mean cell rate of the traffic source (LB_{mean}).

Since a comparison of performance on one network topology is sufficient to show the improvement, the comparison is done for Network_1 only. The load conditions and the network topology are kept the same. The simulator is modified to observe the required variables for comparison. The variable observed for comparison are those that are plotted against load for the two networks in the above section.

6.3.1 Comparison of Average Link Utilizations

Figure 6.11 shows the three plots for the average link utilization versus the average load. Plot 1 shows the average utilization for the network links when the $QLBR$ congestion control scheme is employed. Plots 2 and 3 show the average link utilizations for LB_{peak} and LB_{mean} respectively.

It is clear from the figure that the average link utilization is much better for $QLBR$ as compared to the two versions of LB under all studied load conditions. LB_{mean} has the worst link utilization of the three as it is too conservative. This shows that a preventive congestion control scheme is not efficient for ATM networks. In LB_{mean}, most of the cells (especially of highly bursty traffic sources) are dropped in the source queues due to conservative bandwidth allocation to the sources. LB_{peak} has a better performance than LB_{mean} since most of the cells generated at the sources are transmitted through their first hop. But this fills the transit queues quickly and most of the cells are dropped at the transit queues. $QLBR$ has a very good performance with average link utilization around 0.8. It should be noted that

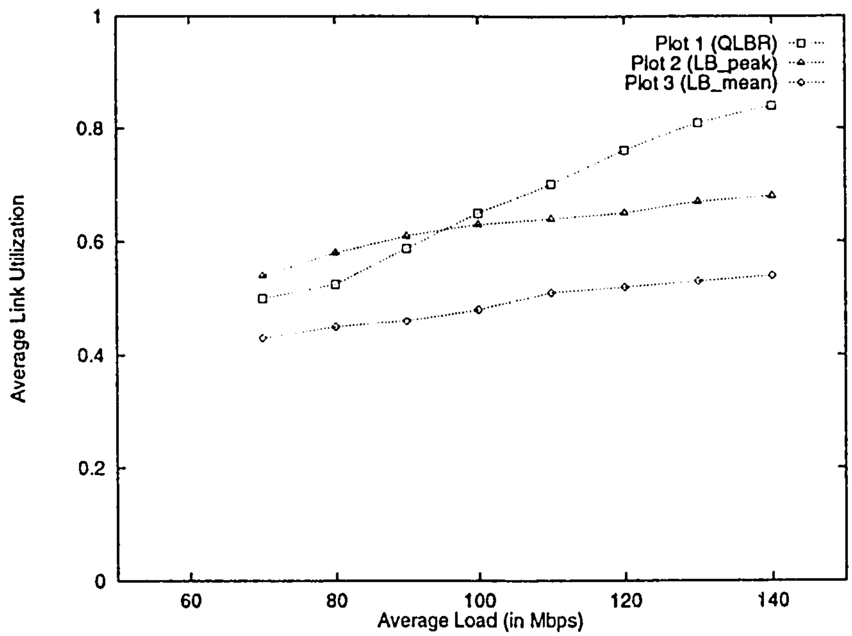

Figure 6.11: Comparison of average link utilizations.

the bit error rate of the links has not been considered in the simulator, which will give a somewhat higher value of link utilization. But the difference is negligible since the value of link bit error rate is around 10^{-15} in optical fibers.

6.3.2 Comparison of Average Delay Jitter

The plots for average delay jitter in $QLBR$, LB_{peak}, and LB_{mean} are shown in Figures 6.12 and 6.13. Plot 1 shows the average delay jitter for the virtual connections when the $QLBR$ congestion control scheme is employed. Plots 2 (Figure 6.12) and 3 (Figure 6.13) show the average delay jitter for LB_{mean} and LB_{peak} respectively.

LB_{mean} gives the best performance for the metric in question. This is clear from

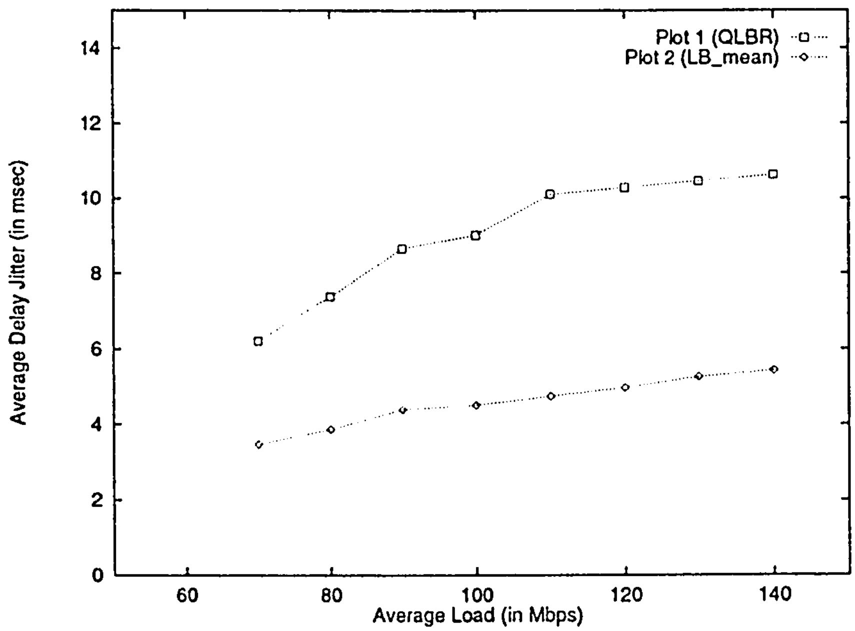

Figure 6.12: Comparison of average delay jitter.

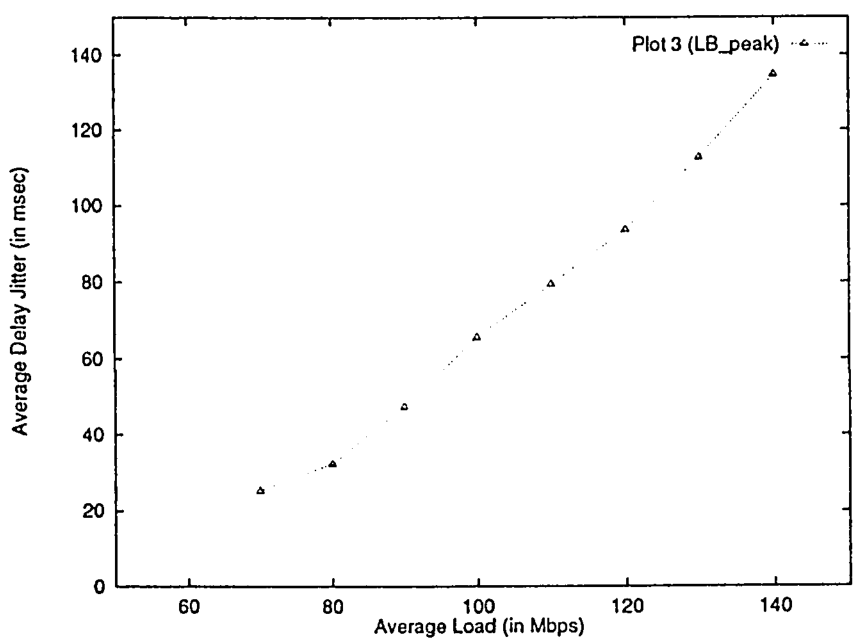

Figure 6.13: Comparison of average delay jitter *(contd).*

Plot 2 in Figure 6.12. The main reason for this better performance of LB_{mean} is the conservative approach of the scheme in accepting cells into the network. At the other end, LB_{peak} has the worst performance of the three schemes, as shown in Figure 6.13. The reason for this poor performance is that almost all the cells generated by the sources are accepted into the network. But most of the cells are dropped after one hop since the transit queues become overloaded. Thus the difference in the time of arrival of two consecutive cells from a source is large.

The average delay jitter for $QLBR$ is somewhat higher than LB_{mean} but its value remains around 10 msec. This is within the acceptable delay tolerance limits of most of the real-time traffic sources [21], for which delay jitter is an important performance metric. The main reason that the delay jitter remains below 10 msec. under varying load conditions is that the congestion thresholds of the transit queues are set according to the delay jitter requirement of the most demanding virtual connection passing through a transit queue.

6.3.3 Comparison of Percentage Cell Loss

Figure 6.14 shows the percentage of the total cells that are lost in the network. The cell loss can be due to the cells dropped at the source queues or the transit queues. Plot 1 shows the percentage cell loss for the $QLBR$ congestion control scheme. Plots 2 and 3 show the this metric for LB_{peak} and LB_{mean} respectively.

$QLBR$ shows a better performance for this metric as compared to both LB_{mean}

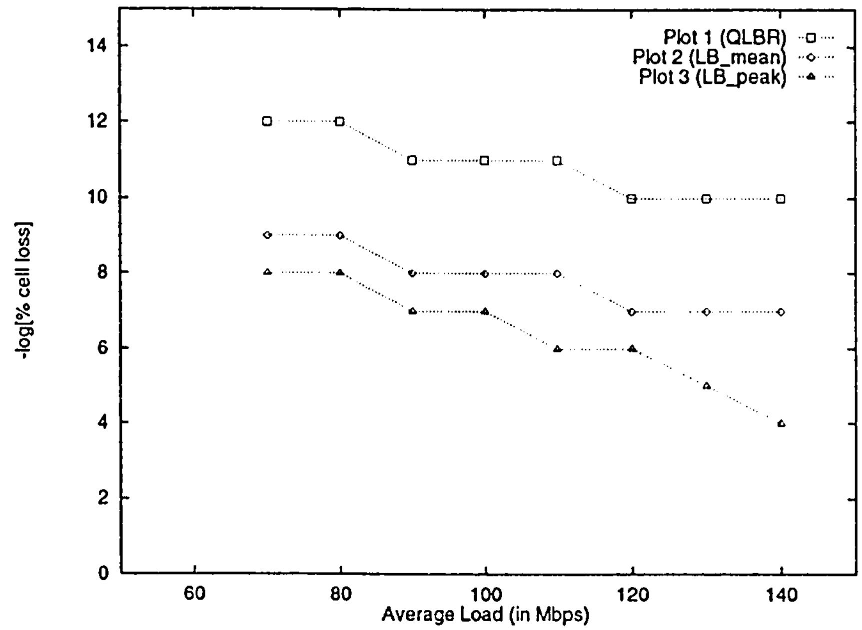

Figure 6.14: Comparison of percentage cell loss.

and LB_{peak} under all load conditions. The percentage cell loss for $QLBR$ ranges from 10^{-12}, for lightly loaded network, to 10^{-10} for heavily loaded network. This is within the cell loss tolerance limits of most of the loss-sensitive traffic sources [21]. The increase in the quantity with increasing load conditions is expected since the probability of cell drop increases with the increase in load.

LB_{mean} (Plot 2) and LB_{peak} (Plot 3) both give a poor performance for this performance metric. Although the performance of LB_{mean} is somewhat better than LB_{peak}, its range (10^{-9} to 10^{-7}) is still higher than the cell loss tolerance of several data sources in ATM networks. LB_{peak} shows even a poorer performance since most of the cells, in this case, are dropped at the transit queues as compared to LB_{mean},

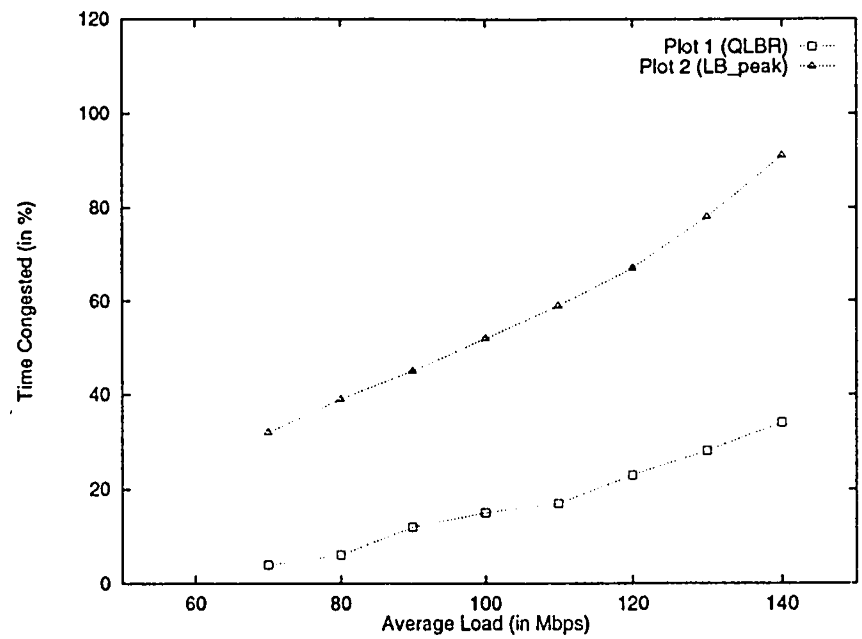

Figure 6.15: Comparison of percentage congested time.

where most of the cells are dropped at the source queues.

6.3.4 Comparison of Percentage Congested Time

Plots 1 and 2 in Figure 6.15 show the percentage of the total time for which one or more of the nodes in the network are in the congested state. Plot 1 shows this metric for $QLBR$ scheme and Plot 2 shows the performance of LB_{peak}. It should be noted here that no congestion takes place in the network when LB_{mean} scheme is used. This is obvious since LB_{mean} is a preventive scheme which does not allow the network to reach the congested state by conservative allocation of bandwidth. That is why no plot for LB_{mean} is shown here.

The percentage congestion time increases for increasing load conditions. But the slope of the curve for LB_{peak} is steeper than that of $QLBR$ for increasing load values. This is because of the multiplicative decrease and additive increase approach used in $QLBR$. Due to this strategy, the network recovers from the congested state quickly and approaches the congested state slowly. But the increase in load does increase the overall average congested time since heavier traffic loads will take the network into the congested state more often than lighter loads.

In comparison, LB_{peak} scheme keeps the network in congested state for longer periods of time. In fact, it can be observed from the figure that even the worst performance of $QLBR$ (at 140 Mbps) is better than the best observed performance of LB_{peak} (at 70 Mbps).

6.4 Discussion

The performance of the proposed scheme is evaluated qualitatively according the the criteria mentioned chapter 1. The performance of the scheme for each criteria is discussed separately.

6.4.1 Efficiency

The efficiency of the scheme can be directly estimated from the efficiency of the network resources utilized. Since memory is not considered as a contented resource in the testing of the scheme, *link utilization* is the only performance metric which can effectively show the efficiency of the network. From the analysis of the average link utilization performed in the above section, it is evident that the scheme keeps the links of the network efficiently utilized.

Therefore, the proposed scheme satisfies the *efficiency* criterion of quality.

6.4.2 Fairness

As defined in chapter 1, *fairness* pertains to the allocation of the resources to the contending clients. In ATM networks, fairness implies the allocation of resources to the traffic sources according to the quality of service (QoS) guaranteed to them by the network at the time of connection establishment.

In the proposed scheme, for simplicity, all the traffic sources are considered to have the same priority level. Under this assumption, the scheme is *fair* since it employs *round robin* discipline of resource (link) allocation to all the contending queues. But, in actual networks, some types of traffic have to be assigned a higher priority than others, e.g., real-time traffic is given a higher priority than data traffic. Assignment of priority levels to different types of traffic and the allocation of resources according to these priorities can be a future extension to the proposed scheme.

Another aspect of fairness is the penalizing of only those sources which are the cause of congestion in the network. From the mechanism of the proposed scheme, it can be seen that when a node is congested, the traffic sources attached directly to that node are the first to reduce their token rates. Its immediate predecessor nodes are effected next, although, the sources which are the main cause of congestion might be further upstream. In this respect, the scheme is not fair, but, since the token rates are never reduced below the average cell rate of a source, the major aspect of fairness is still satisfied.

6.4.3 Distributedness

The *distributedness* criterion is especially important for high speed networks. Centralized congestion control mechanisms require that the information about the state of the network is gathered at one point which is solely responsible for taking any decisions with respect to the congested state. But the large propagation delays in ATM networks makes this approach impractical. Therefore, the congestion control in these networks should be distributed so that the nodes affected by the congested state can take the appropriate measures quickly.

Our scheme satisfies this criterion safely since each node is independent in reacting to the congested state. Furthermore, the information about the congested state is kept within the close neighborhood of one hop. This is important because the propagation of this information farther away might affect the network adversely due to its relatively long propagation delays.

6.4.4 Convergence

Responsiveness and *Smoothness* are the two important sub-criteria of *Convergence.* Both these sub-criteria should be satisfied in order for a congestion control mechanism to converge to an equilibrium point. The satisfaction of the convergence criterion guarantees that the mechanism will keep the network from collapsing due to congestion.

The responsiveness sub-criterion is satisfied by the proposed scheme because the reaction to the congested state is immediate. Moreover, the congestion state is *anticipated* by the size of the transit queues, which makes the responsiveness of the proposed scheme even better. Figure 6.16 shows a trace curve of a transit queue[1] with congestion threshold set equal to 100 cells. It can be seen from the figure that the reaction to the congested state is swift because of multiplicative decrease strategy.

From Figure 6.16 it can also be observed that the increase in the size of the queue start taking place when it becomes empty. This is because the congestion relief threshold for the queue is set at queue size equal to 0. But the increase in the size of the queue is gradual as compared to the decrease at congestion. The additive increase strategy is the main reason for this trend. Overall, the curve is not very smooth but it does show a convergence around an equilibrium point. It should be noted that as much as 300,000 or more cells may get transmitted in one second.

[1]It is not necessary that every transit queue has a similar trace curve.

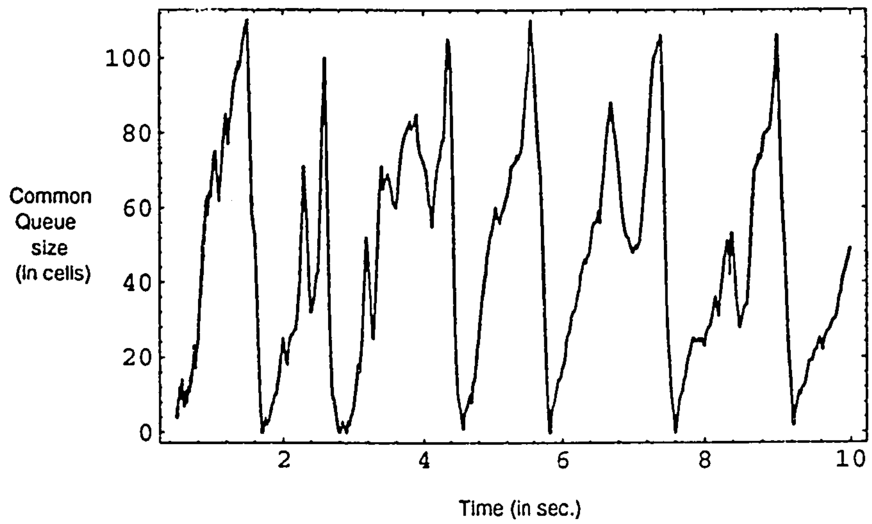

Figure 6.16: Trace plot of a Transit queue for 10 sec.

Hence, the lack of smoothness may be attributed to the time scale of the plot.

The multiplicative increase and additive increase policy has been proved to be optimal for congestion control mechanisms [10]. To improve the smoothness of the proposed scheme, the choice of the multiplicative and additive factor have to be further investigated. This can be performed as a future extension to this work.

Chapter 7

Conclusions

In this thesis, a new congestion control mechanism for ATM networks, based on the *reactive* congestion control principle, is designed and evaluated. A simulator was implemented to evaluate the performance of the proposed scheme under a variety of traffic loads.

A brief introduction to the ATM and traffic control was presented in Chapter 1. Important terminology related to traffic control, in general, and ATM, in particular, was also described therein. Traffic source models is an important aspect in the design of a congestion control scheme for ATM networks. Chapter 2 described the different traffic source models developed in the literature for ATM networks. A comprehensive survey of previous work performed in the area of congestion control for ATM networks was given in Chapter 3. Chapter 4 described the methodology employed in the design of the new scheme. It presented the estimation of the important parameters. Chapter 5 described the simulator including the input parameters required by the simulator and the output variables reported by it. Chapter 6 presented the results of important performance metrics for two network topologies under different load conditions. These results were then compared with those obtained using a simple leaky bucket congestion control mechanism.

From the simulation results it can be concluded that the scheme shows good performance under various load conditions. The cell loss in the tested network topologies show that the cell loss in these networks is within the limits usually required in high speed networks supporting real-time traffic. The cell delay jitter in the networks is also within the expected limit for moderately loaded networks. But it becomes a little higher for some demanding real-time applications when the network is heavily loaded. This is mainly because of no priority distinction for traffic in the *Transit queues.* The link utilization for the scheme is also good.

The comparison with simple leaky bucket scheme shows that the proposed scheme has a superior performance. The improvement in performance is mainly due to its localized feedback approach which does not have the long delay of end-to-end feedback approach. The low processing requirement of the scheme also contributes to its better results.

7.1 Future Work

The problem of congestion control is so complex that there is always room for further improvements. In this section, we suggest some of the possible enhancements to the proposed scheme which, in our opinion, can further improve the performance of the proposed scheme. Some suggestions to enhance the simulator are also proposed in this section.

7.1.1 Improvements to the *QLBR* Scheme

The following improvements are suggested to enhance the performance of the scheme.

- Division of transit queue into multiple queues with different priority levels.
- Controlling the output rate of the transit queue upon congestion notification from downstream node.
- Prioritizing the *Source* queues based upon the type of source.
- Using statistical measurements and methods to dynamically change the *Congestion* and *Relief* thresholds of the transit queues.

7.1.2 Enhancements to the Simulator

Some of the improvements that can be made in the simulator are as follows.

- Developing a graphical user interface to make the simulator user friendly.

- Employing object-oriented paradigm so that addition of new functions to the simulator becomes easy.
- Developing a distributed simulator to reduce the simulation time.
- Using different traffic models to effectively model different types of traffic sources.

Appendix A

This appendix contains the flowcharts of the important routines used in the simulator. The simulator is designed as a discrete event simulator with five major types of events. The flowcharts for these five types of events are presented below. A brief description about each type of event also follows.

Type 1: Token Generation

This event generates the tokens for each token pool attached to a traffic source at its source node. The tokens are generated according to the token generation rate of the source at the time the event takes place. Figures A.2 and A.3 show the flowcharts for this event.

Type 2: Cell Generation

This event generates cells for a traffic source according to its specific parameters. The cells are generated according to the ON-OFF traffic model, i.e., each burst of cells is followed by an idle period. At the start of each ON period, the burst size is calculated. Upon each occurance of cell generation event, one cell is generated and put in the corresponding source queue (if it is not full). At the end of the burst, OFF period is generated and the next burst is scheduled after this period. Figures A.4 and A.5 show the flowcharts for this event.

Type 3: Cell Transmission

This event transmits cells from a queue to the next node. The transmission takes place in a round-robin fashion. The multiplexer routine shown in the flowchart of Figure A.6 performs the round-robin allocation of the link to the queues for transmission. Figures A.7 and A.8 show the flowchart of the routine which simulates the transmission from a *source* queue. The routine also schedules the next transmission and the arrival of the cell at the destination node. Figure A.9 shows the flowchart of a similar routine for the *transit* queue.

Type 4: Cell Arrival

Two type of cells can arrive at any node in the simulator. *Normal* cells are those which are generated by a traffic source, and *Notification* cells which are send to indicate the start or end of the congestion at a neighboring node. Separate routines are required for each type of cell. Figures A.10 and A.11 show the flowchart for processing the arrival of a normal cell at a node. This includes indentifying the next node for the cell (if current node is not its destination) and insertion of the cell in the appropriate transit queue (if it is not full).

The arrival of a notification cell is processed by a routine which schedules the token rate increase or decrease event corresponding to the type of notification cell. Figure A.12 shows the flowchart for this routine.

Type 5: Token Rate Change

Token rate change is performed relative to the state of congestion at a node itself and at its immediate neighbors. Corresponding to this state, the token rate can be either increased or decreased. Since this change is performed periodically in the scheme, the token rate change event is scheduled in the same manner. Figure A.13 shows the flowchart of the routine for this type of event.

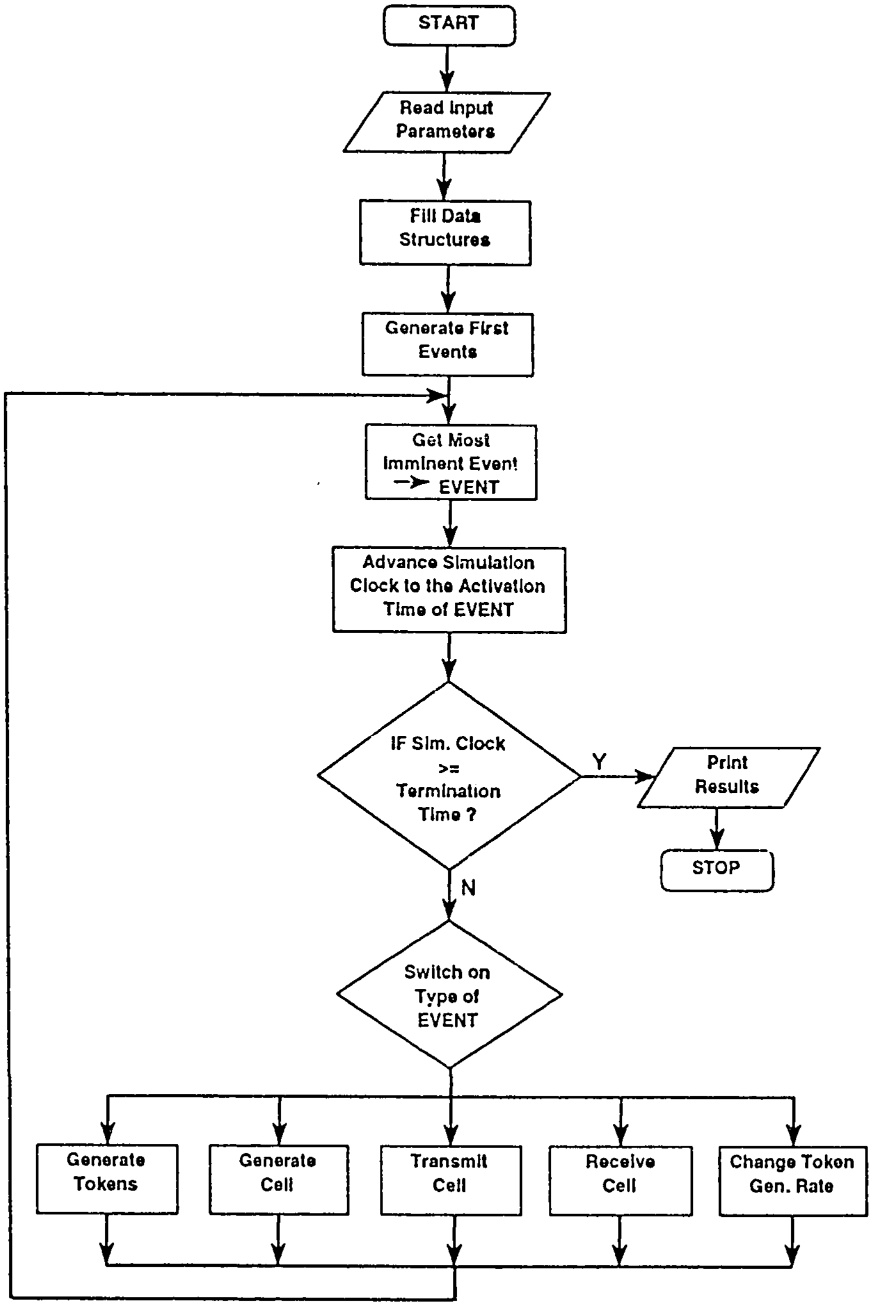

Figure A.1: Main routine.

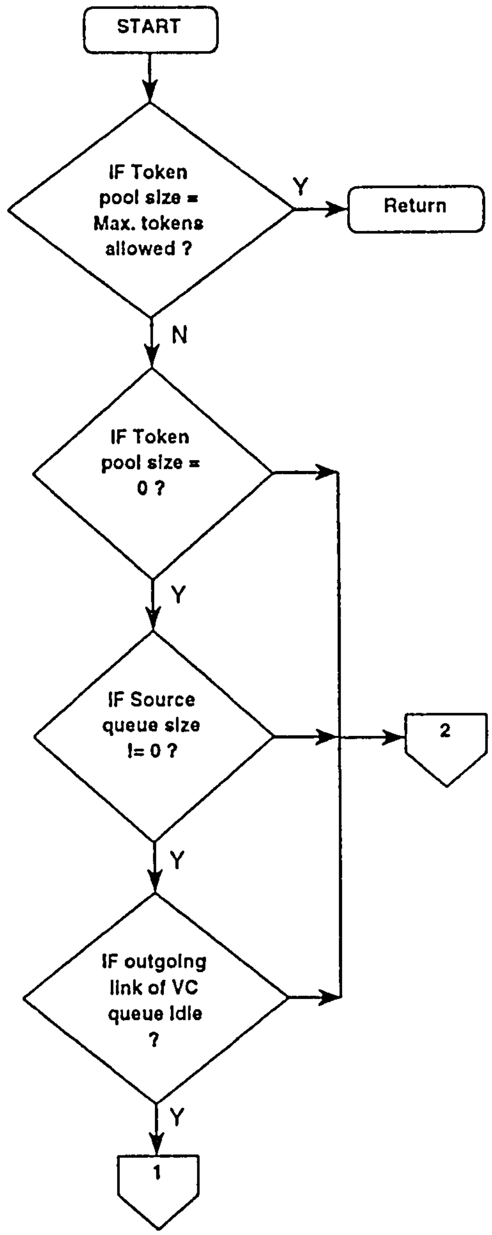

Figure A.2: Token generation routine.

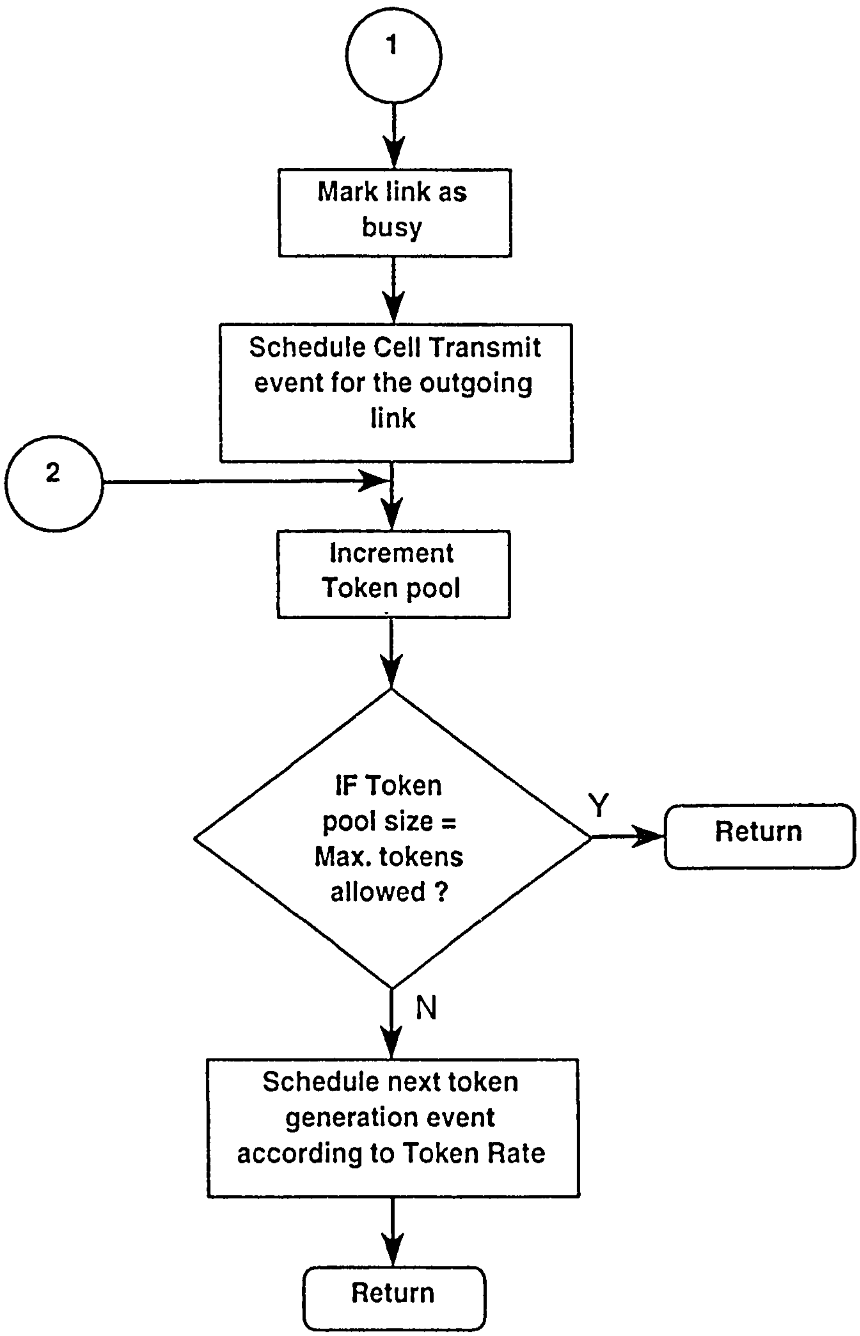

Figure A.3: Token generation routine *(contd.)*.

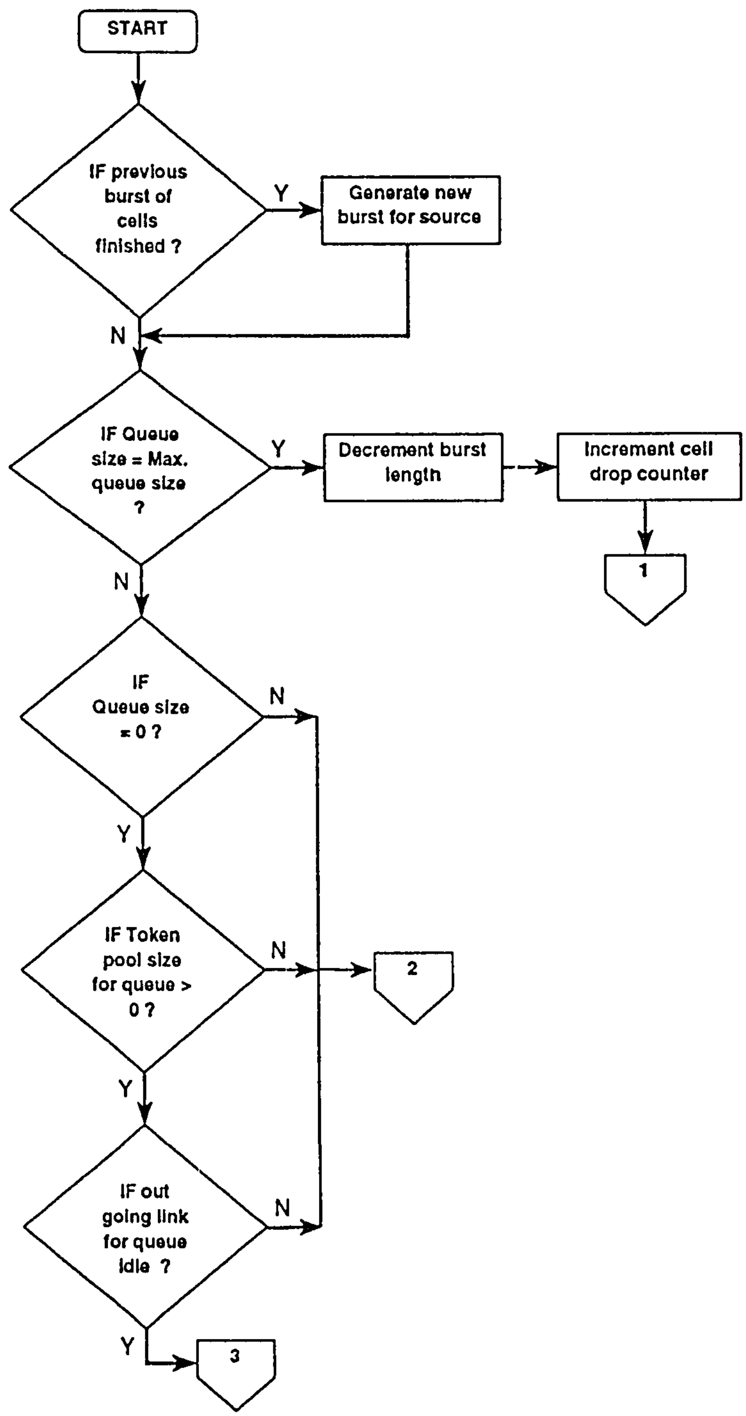

Figure A.4: Cell generation routine.

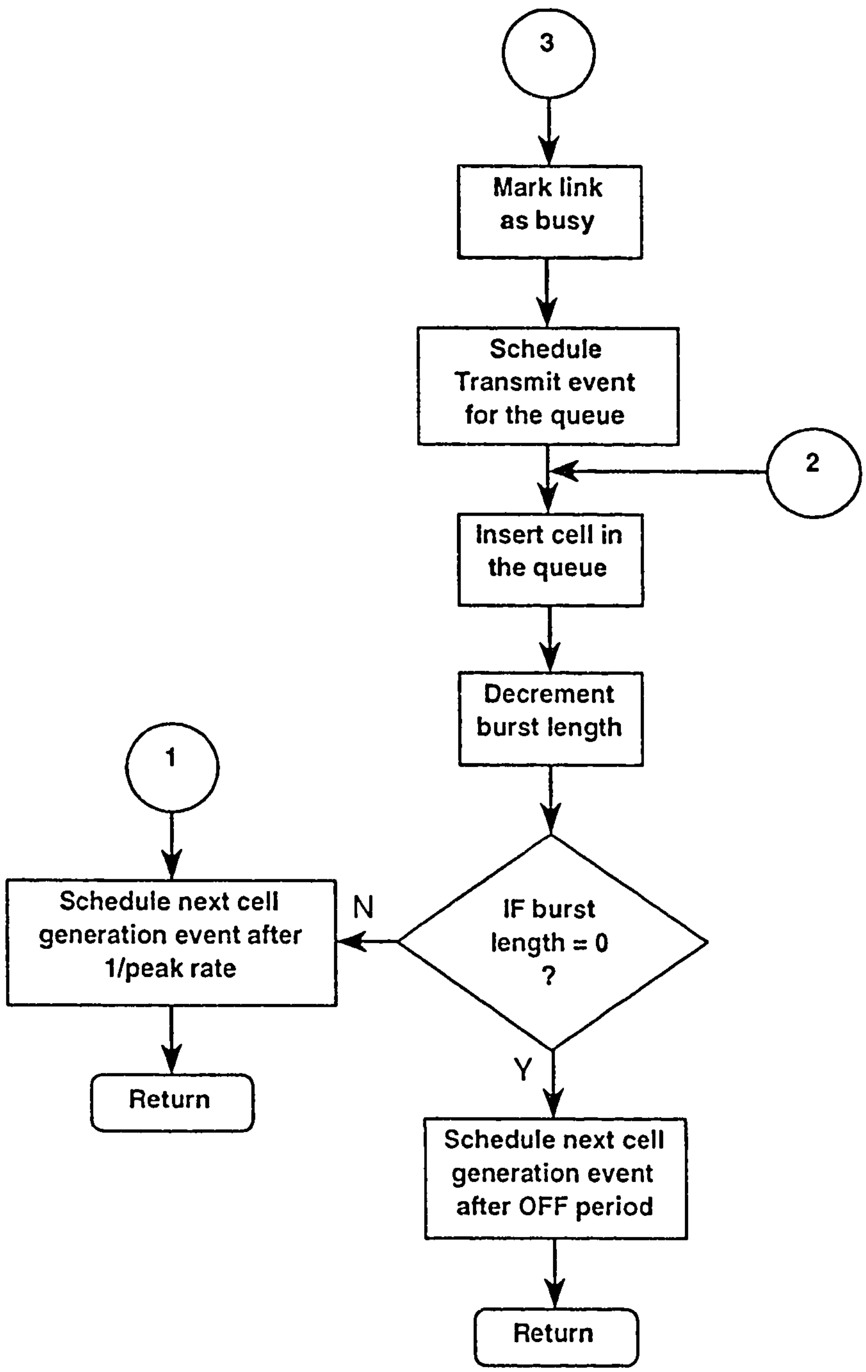

Figure A.5: Cell generation routine *(contd.)*.

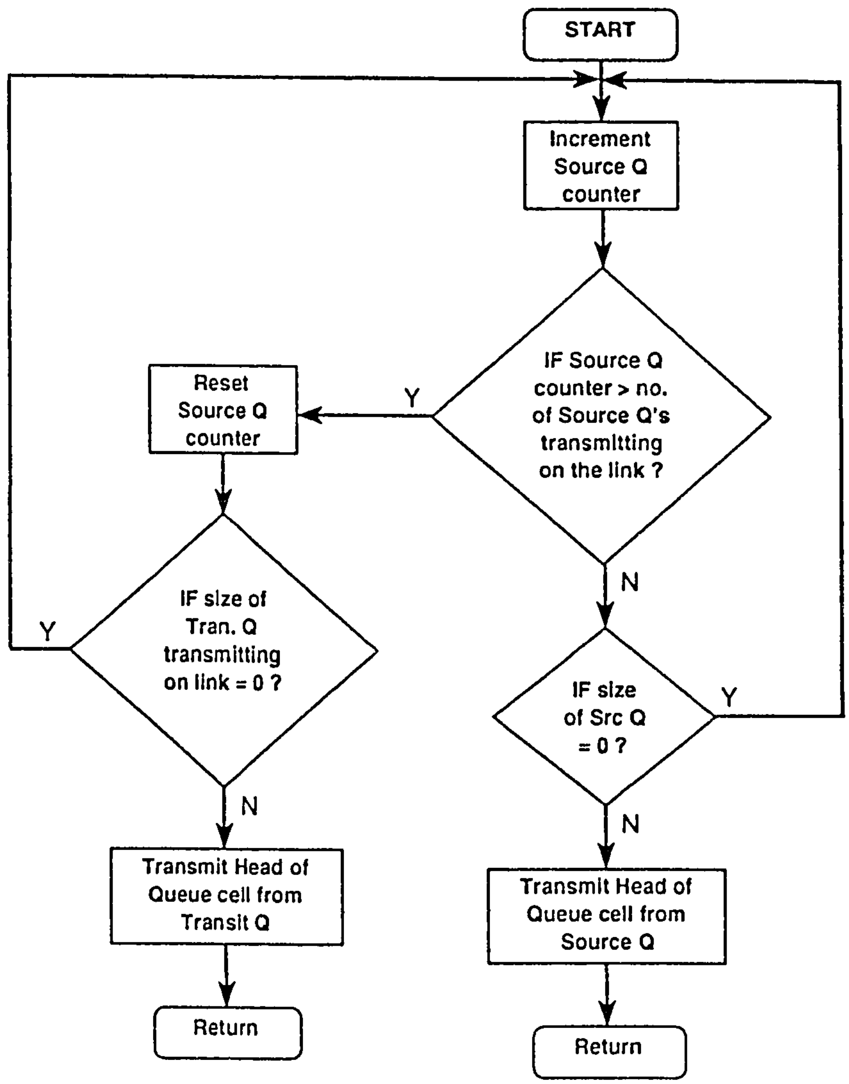

Figure A.6: Multiplexer routine.

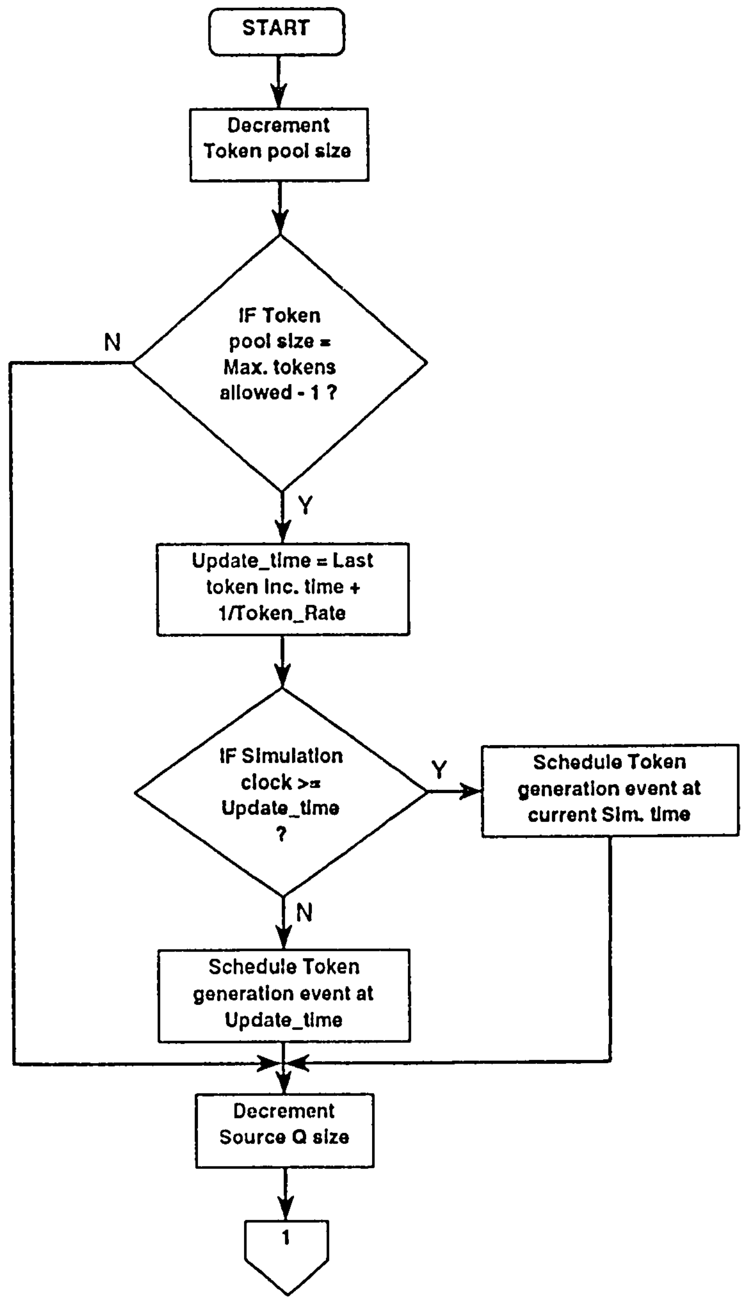

Figure A.7: Routine of cell transmission from source queue.

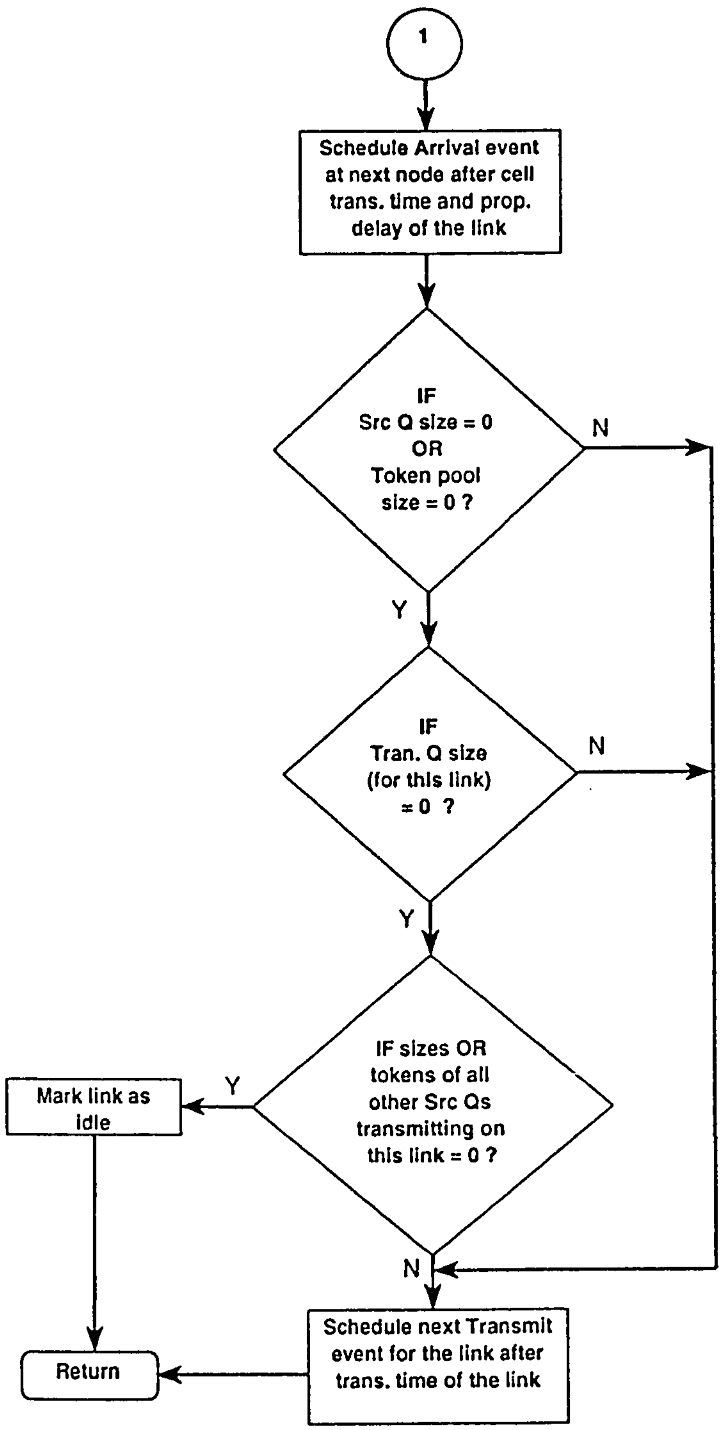

Figure A.8: Routine of cell transmission from source queue *(contd.)*.

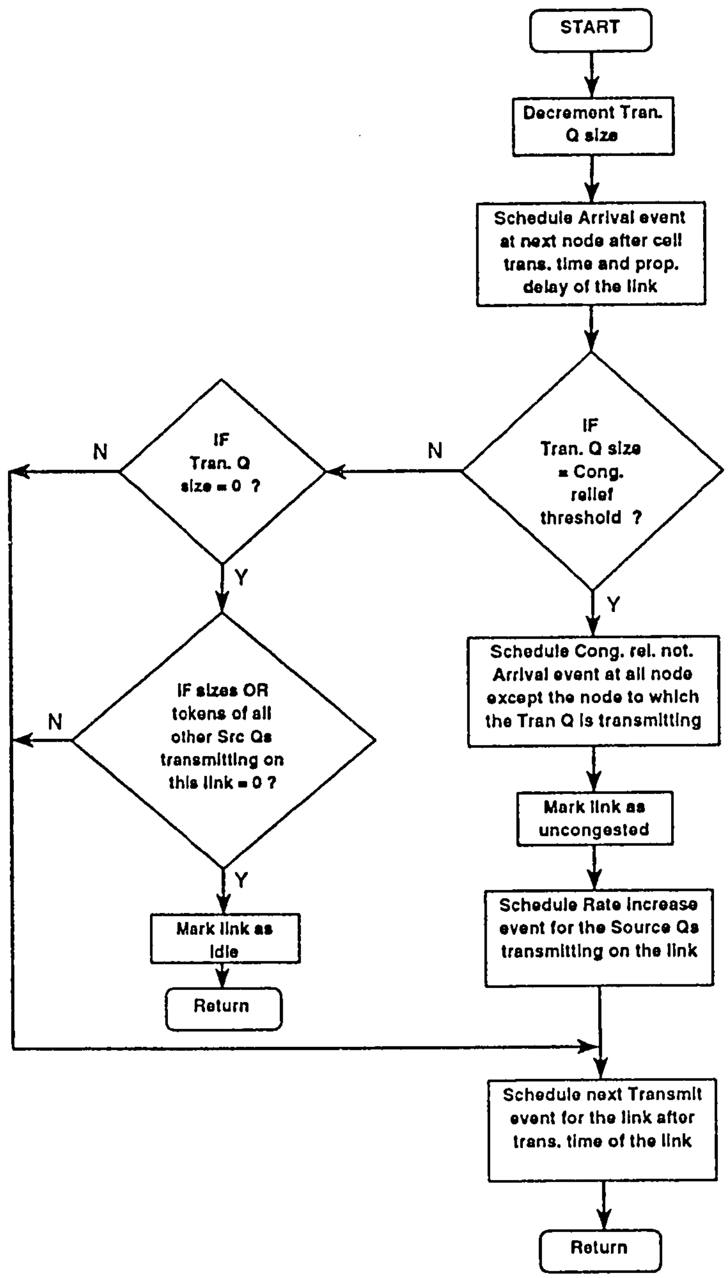

Figure A.9: Routine of cell transmission from transit queue.

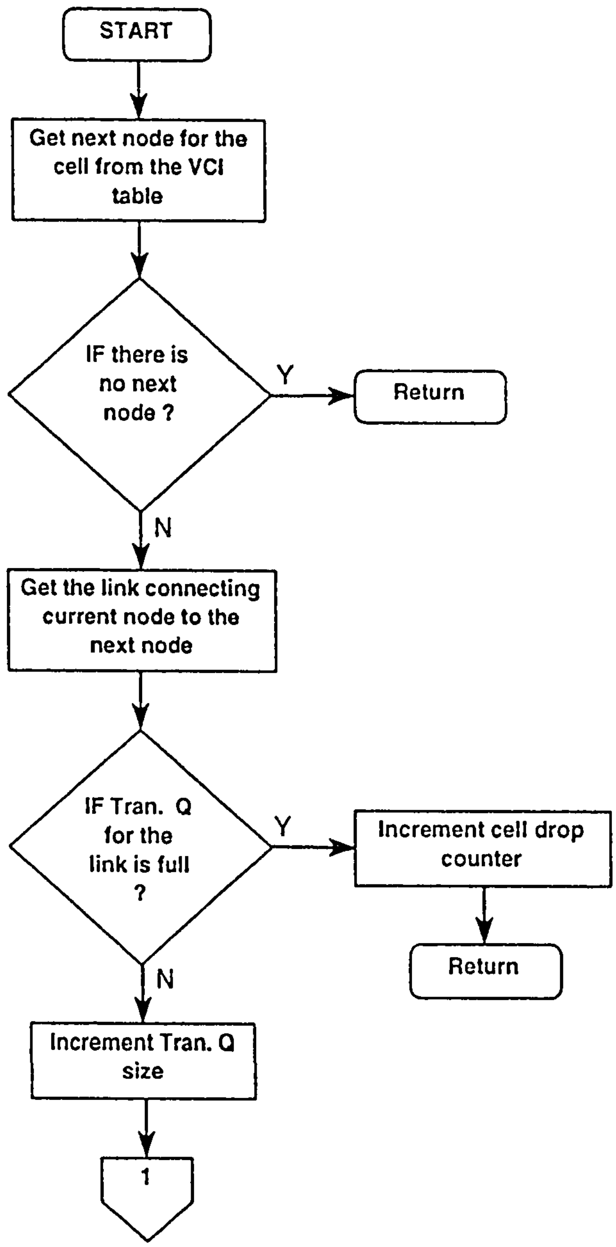

Figure A.10: Routine for processing the arrival of normal cell.

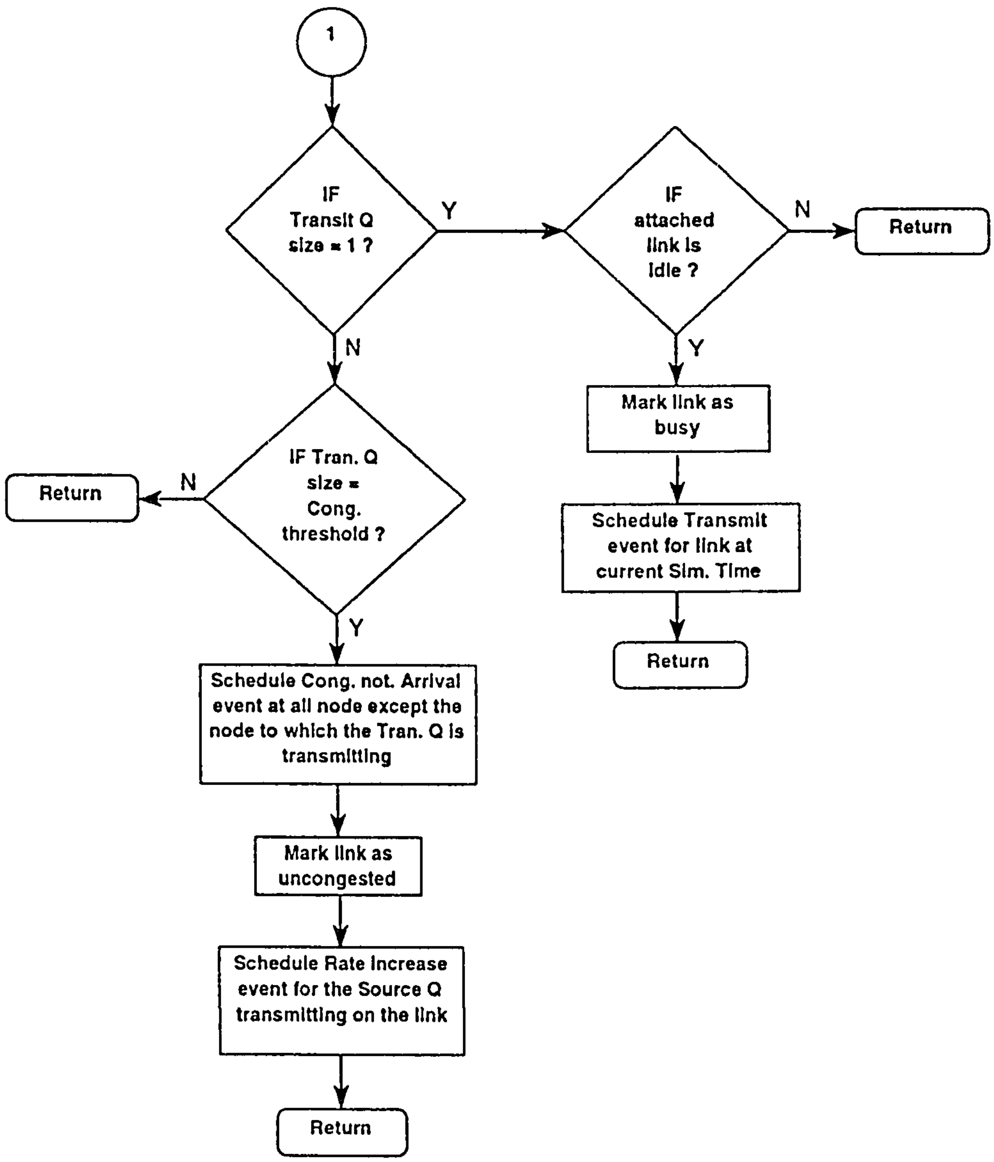

Figure A.11: Routine for processing the arrival of normal cell *(contd.)*.

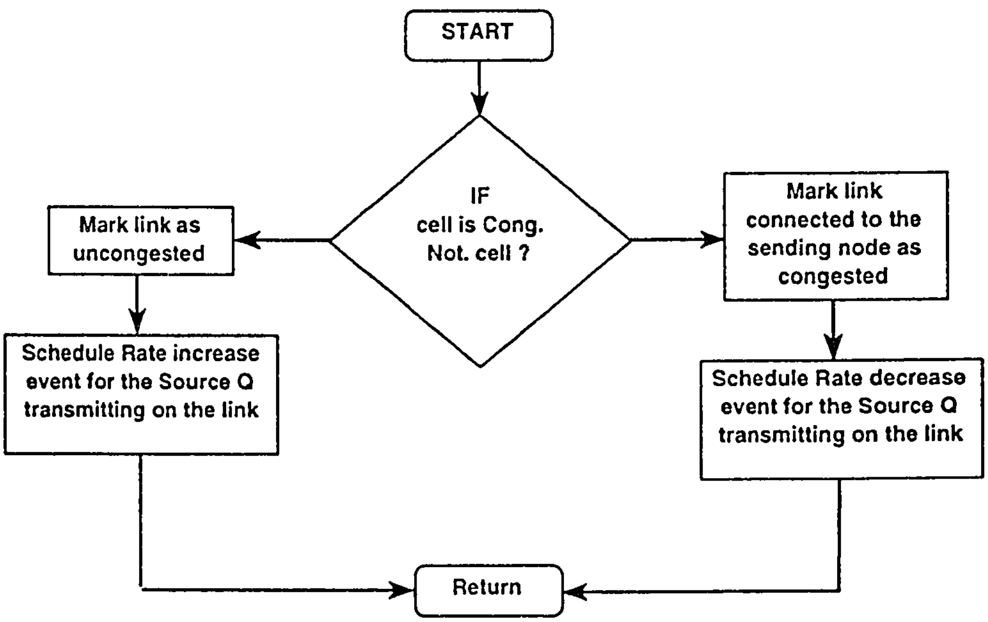

Figure A.12: Routine for processing the arrival of notification cell.

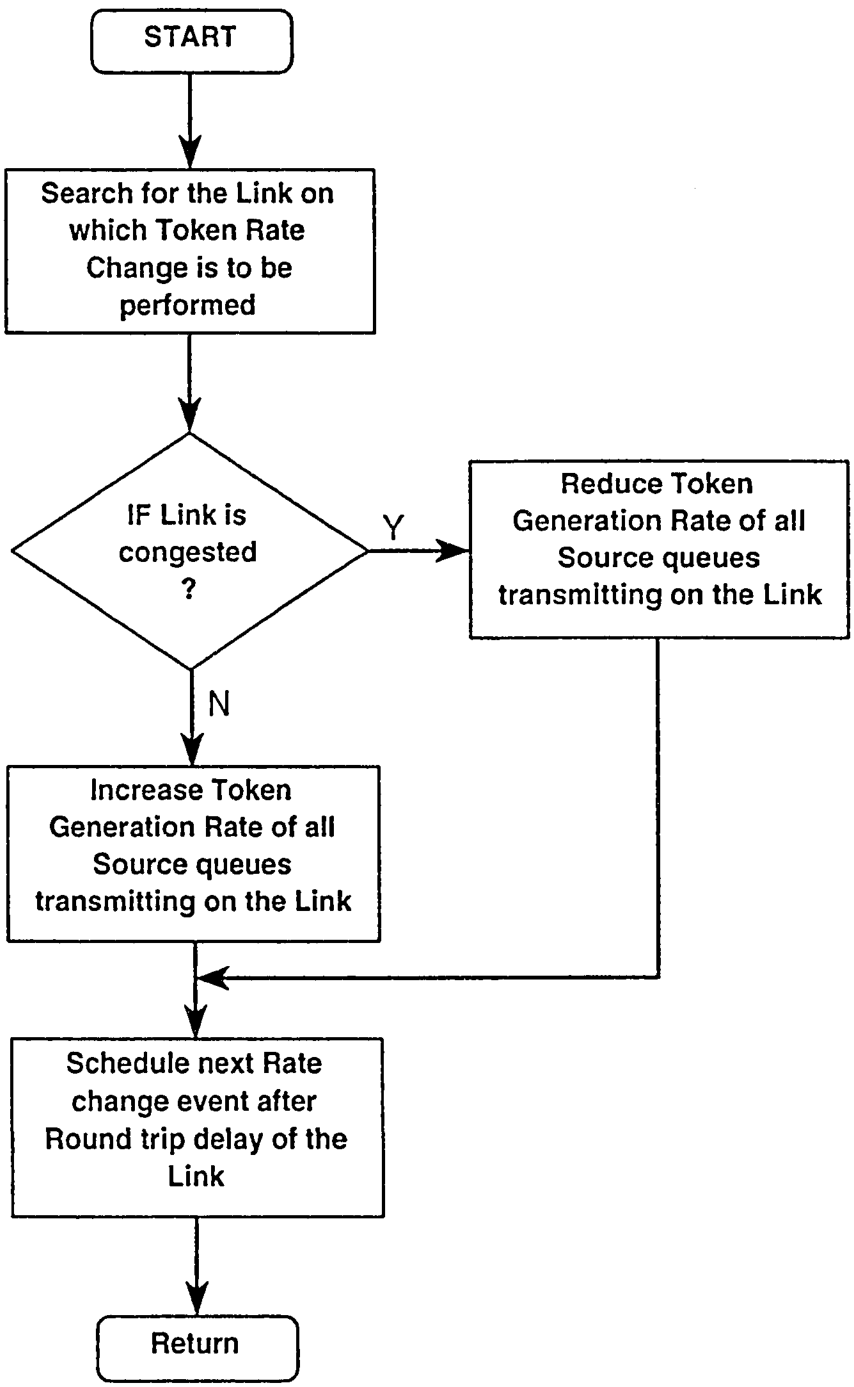

Figure A.13: Token rate change routine.

Bibliography

[1] D. Bertsekas and R. Gallager. *Data Networks, 2nd Edition.* Prentice-Hall, Englewood Cliffs, N.J., 1992.

[2] R. Händel and M. N. Huber. *Integrated Broadband Networks.* Addison-Wesley Wokingham, England., 1991.

[3] CCITT, Geneva. *Recommendation I.121: 'Broadband Aspects of ISDN'*, 1989.

[4] CCITT, Geneva. *Recommendation I.113: 'Vocabulary of Terms for Broadband Aspects of ISDN'*, 1991.

[5] ATM Forum. *ATM Forum: 'ATM User-Network Interface Specification'*, 1992.

[6] CCITT, Geneva. *Recommendation I.321: 'B-ISDN Protocol Reference Model and its Applications'*, 1991.

[7] M. de Prycker. *Asynchronous Transfer Mode: solution for broadband ISDN.* Ellis Horwood Ltd., England, 1991.

[8] CCITT, Geneva. *Recommendation I.320: 'ISDN Protocol Reference Model'*, 1989.

[9] CCITT, Geneva. *Recommendation I.361: 'B-ISDN ATM Layer Specification'*, 1991.

[10] D-M Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, (17):1–14, 1989.

[11] A. S. Tanenbaum. *Computer Networks, 2nd Edition.* Prentice-Hall, Englewood Cliffs, N.J., 1988.

[12] B. Field and T. Znati. α-Channel: A network framework to support application real-time performance guarantees. *IEEE J. on Selected Areas in Communications*, 11(8):1317–1329, Oct. 1993.

[13] L. Goldberg. ATM's growing pains bring maturity as UNI 4.0 evolves. *Electronics Design*, pages 51–60, May 30 1995.

[14] A. E. Eckberg. Generalized peakedness of teletraffic processes. In *10th International Teletraffic Conf.*, 1983.

[15] K. Fendick and W. Whitt. Measurements and approximations to describe the offered traffic and predict the average workload in a single-server queue. *Proceedings of the IEEE*, 77:171–194, 1989.

[16] H. Heffes and D. M. Lucantoni. A Markov-Modulated characterization of packetized voice and data and related statistical multiplexer performance. *IEEE J. Selected Areas Comm.*, SAC-4:856–886, 1986.

[17] G. Ramamurthy and B. Sengupta. Modeling and analysis of a variable bit rate video multiplexer, 1992.

[18] D. Heyman, A. Tabatabai, and T. V. Lakshman. Statistical analysis and simulation study of video teletraffic in ATM networks. *IEEE Trans. Circuits and Systems for Video Technology*, 2:49–59, 1992.

[19] W. E. Leland, W. Willinger, M. S. Taqqu, and D. V. Wilson. On the Self-Similar nature of Ethernet traffic. *Computer Communication Review*, pages 203–213, 1995.

[20] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Variable-Bit-Rate video traffic and long range dependence. *IEEE Transactions on Communications*, 1992.

[21] K. DuBose and H. S. Kim. An effective bit rate/table lookup based admission control algorithm for the ATM B-ISDN. In *17th Conference on Local Computer Networks*, pages 20–29. IEEE, 1992.

[22] J. J. Bae and T. Suda. Survey of traffic control schemes and protocols in ATM networks. *Proceedings of the IEEE*, 79(2):170–189, Feb. 1991.

[23] C. A. Cooper and K.I. Park. Toward a broadband congestion control strategy. *IEEE Network Magazine*, pages 18–23, May 1990.

[24] M. Decina, T. Toniatti, P. Vaccari, and L. Verri. Bandwidth assignment and virtual call blocking in ATM networks. In *INFOCOM '90*. IEEE, June 1990.

[25] G. Gallassi, G. Rigolio, and L. Verri. Resource management and dimensioning in ATM networks. *IEEE Network Magazine*, pages 8–17, May 1990.

[26] H. Suzuki, T. Murase, S. Sato, and T. Takeuchi. A burst traffic control strategy for ATM networks. In *GLOBECOM '90*. IEEE, Dec. 1990.

[27] G. M. Woodruff and R. Kositpaiboon. Multimedia traffic management principles for guaranteed ATM network performance. *IEEE J. Selected Areas Comm.*, 8(3):437–446, 1990.

[28] H. Esaki, K. Iwamura, and T. Kodama. A simple and effective admission control method for an ATM network. In *GLOBECOM '90*. IEEE, Dec. 1990.

[29] T. R. Griffiths. Analysis of connection acceptence strategies for Asynchronous Transfer Mode networks. In *GLOBECOM '90*. IEEE, Dec. 1990.

[30] B. Jabbari. A bandwidth allocation technique for high speed networks. In *GLOBECOM '90*. IEEE, Dec. 1990.

[31] G. Ramamurthy and R. S. Dighe. Distributed source control: a network access control for integrated broadband packet networks. *IEEE J. Selected Areas Comm.*, 9(7):990–1001, 1991.

[32] K. Y. Siu and H. Y. Tzeng. Adaptive proportional rate control for ABR service in ATM networks. In *14th International Phoenix Conf. on Computers and Communications*, pages 529–535. IEEE, 1995.

[33] A. Gersht and K. J. Lee. A congestion control framework for ATM networks. In *INFOCOM '89*, pages 701–711. IEEE, 1989.

[34] S. J. Golestani. Congestion-free transmission in high speed packet networks. *IEEE Trans. Comm.*, 39(12):1802–1812, 1991.

[35] L. Zhang. Virtual clock: a new traffic control algorithm for packet switching networks. In *SIGCOMM '90*. ACM, Sept. 1990.

[36] W. Matragi and K. Sohraby. Combined open-loop/closed-loop approach for congestion control in ATM networks. In *ICC '93 Geneva*, pages 1336–1342. IEEE, 1993.

[37] S. Saunders. ATM Forum ponders congestion control options. *Data Communications*, pages 55–60, March 1994.

[38] D. Hong and T. Suda. Congestion control and prevention in ATM networks. *IEEE Network Magazine*, pages 10–16, July 1991.

[39] I. Chlamtac and T. Zhang. A counter based congestion control (CBC) for ATM networks. *Computer Networks and ISDN Systems*, 26:5–27, 1993.

[40] E. D. Sykas, K. M. Vlakos, I. Ch Paschalidis, and G K Mourtzinou. Congestion avoidance for ATM networks. *Computer Communications*, 17(9):657–662, Sept. 1994.

[41] J. M. Peha. The priority token bank: Integrated scheduling and admission control for an integrated-services network. In *ICC '93 Geneva*, pages 345–351. IEEE, 1993.

[42] G. L. Li. Admission control for transmitting connectionless traffic across ATM networks. In *14th International Phoenix Conf. on Computers and Communications*, pages 543–549. IEEE, 1995.

[43] S. Abe and T. Soumiya. A traffic control method for service quality assurance in an ATM network. *IEEE J. on Selected Areas in Communications*, 12(2):322–331, Feb. 1994.

[44] S. Chatterjee and M. A. Bassiouni. Dynamic congestion control schemes in high-speed multimedia networks. In *18th Conf. on Local Computer Networks*, pages 81–90. IEEE, 1993.

[45] Y. H. Jeon and I. Viniotis. Achievable loss probabilities and bufffer allocation policies in ATM nodes with correlated arrivals. In *ICC '93 Geneva*, pages 365–369. IEEE, 1993.

[46] L. G. Mason, Z. Dziong, and N. Tetreault. Fair-efficient call admission control policies for broadband networks. In *ICC '93 Geneva*, pages 976–982. IEEE, 1993.

[47] A. Raha, N. Malcolm, and W. Zhao. Performance evaluation of admission policies in ATM based embedded real-time systems. In *19th Conf. on Local Computer Networks*, pages 129–138. IEEE, Oct. 1994.

[48] A. Hiramatsu. ATM communication network control by neural networks. In *Intl. Conf. on Neural Networks*, June 1989.

[49] B. Khasnabish. Integrated bandwidth and congestion management in ATM networks using a simple learning algorithm and a bucket bank. In *18th Conf. on Local Computer Networks*, pages 388–396, 1993.

[50] L. K. Reiss and L. F. Merakos. Priority shaping of source traffic in ATM B-ISDN. *Computer Communications*, 16(12):794–797, Dec. 1993.

[51] F. Bernabai, L. Gratta, M. Listanti, and M. Testa. Analysis of two level shaping for multiplexing of ON-OFF ATM sources. In *ICC '93 Geneva*, pages 1380–1385. IEEE, 1993.

[52] K. Sohraby and M. Sidi. On the performance of bursty and modulated sources subject to leaky bucket rate-based access control schemes. *IEEE Transactions on Communications*, 42:477–487, Feb/Mar/April 1994.

[53] E. P. Rathgeb. Modeling and performance comparison of policing mechanisms for ATM networks. *IEEE J. on Selected Areas in Communications*, 9(3):335–342, April 1991.

[54] M. Buttó, E. Cavallero, and A. Tonietti. Effectiveness of the "Leaky Bucket" policing mechanism in ATM networks. *IEEE J. on Selected Areas in Communications*, 9(3):335–342, April 1991.

[55] Y. J. Kim, B. C. Shin, and C. K. Un. Performance analysis of leaky-bucket bandwidth enforcement strategy for bursty traffics in an ATM network. *Computer Networks and ISDN Systems*, 25:295–303, 1992.

[56] D. K. Hsing. Performance study on the "Leaky Bucket" usage parameter control mechanism with CLP tagging. In *ICC '93 Geneva*, pages 359–364. IEEE, 1993.

[57] L. Dittmann, S. B. Jacobsen, and K. Moth. Flow enforcement algorithms for ATM networks. *IEEE J. on Selected Areas in Communications*, 9(3):343–350, April 1991.

[58] N. Yin, San-Qi Li, and T.E. Stern. Congestion control for packet voice by selective packet discarding. *IEEE Transactions on Communications*, 38(5):674–683, May 1990.

[59] R. Jain and K. K. Ramakrishnan. Congestion avoidance in computer networks with a connectionless network layer: Concepts, goals and methodology. In *Computer Networking Symposium*, pages 134–143, April 1988.

[60] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis.* McGraw Hill, Inc., 1991.

Vita

- Sabih Masarrat Alsayed
- Received Bachelor's degree in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan in August, 1991.
- Worked as Hardware Engineer in the Engineering Services Group of M/s LANTECH Systems (Pvt.) Ltd., a networking firm based in Islamabad, from Jan. till Mar. 1992 .
- Received advanced training in computer systems at Computer Training Center, Islamabad from May 1992 to Jan. 1993.
- Completed Master's degree requirements at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in July, 1996.